

Feature Extraction from Remotely Sensed Imagery
using
Artificial Neural Networks

*A thesis submitted
in partial fulfillment of the requirements
for the degree of*

MASTER OF TECHNOLOGY

by

Akshay Kant

to the

**DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
March, 1996**

10 APR 1996

CENTRAL LIBRARY

111 1st PUR

121271
ISS. No. A.

CE-1996-M-KAN-FEA



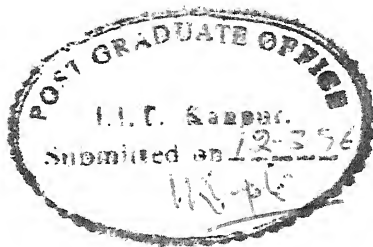
A121271

CERTIFICATE

It is certified that the work presented in this thesis entitled **“FEATURE EXTRACTION FROM REMOTELY SENSED IMAGERY USING ARTIFICIAL NEURAL NETWORKS”** has been carried out by Shri Akshay Kant (Roll No.9410301) under my supervision and has not been submitted elsewhere for a degree.

March 12, 1996

Nitin Tripathi
(Dr. Nitin Kumar Tripathi)
Assistant Professor
Dept. of Civil Engineering
Indian Institute of Technology
Kanpur-208 016



DEDICATED

TO

MY PARENTS

ACKNOWLEDGEMENT

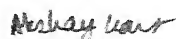
I express, with all my heart, sincere gratitude and reverence for my Guru Dr. N. K. Tripathi, without whose expert guidance, this work was not possible. His ever helping attitude and extremely encouraging remarks made me feel at home at I.I.T. Kanpur. It has been a great pleasure to be with him and I would cherish these lovely moments throughout my life. I, therefore, dedicate all my successful endeavors in I.I.T. Kanpur to Dr. Tripathi.

I am also thankful to Dr. O. Dixit for his valuable suggestions during my work.

I am extremely thankful to Dr. C.V.R. Murthy for his help.

Thanks are due to Mr. G. P. Mishra Mr. Ramkishan for their co-operation throughout my stay at I.I.T. Kanpur.

I wish to thank all my friends who have made my stay in this institute a comfortable one. In this regard I am, especially, thankful to Ajai Srivastava, Bharat Chaturvedi, Gentleman, Desh, Ritwik, Paro, Prakash Veer, Sandp and Chacha Rai for their help during my thesis work.


Akshay Kant

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

CHAPTER

| | |
|--|------------|
| 1. INTRODUCTION | 1-1 |
| 1.1 General | 1-1 |
| 1.2 Study Area | 1-4 |
| 1.2.1 Singrauli coal basin | 1-4 |
| 1.2.2 Semi urban environment of Kanpur | 1-7 |
| 1.3 Data Acquisition | 1-9 |
| 1.3.1 Singrauli Area | 1-9 |
| 1.3.2 Semi urban environment of Kanpur | 1-9 |
| 1.4 Computing Platforms for the Study | 1-10 |
| 1.4.1 Hardware used | 1-10 |
| 1.4.2 Software details | 1-10 |
| 1.5 Scope of the work | 1-10 |
| 1.6 Organization of the work | 1-11 |

| | |
|---|----------------|
| 2. Literature review | 2-1 |
| 2.1 Image Classification | 2-1 |
| 2.1.1 Supervised classification | 2-2 |
| 2.1.2 Unsupervised classification | 2-3 |
| 2.2 Overview of the earlier work | 2-3 |
| 3. Statistical Pattern Recognition | 3-1 |
| 3.1 Introduction | 3-1 |
| 3.2 Supervised Classification using baye's maximum likelihood classification | 3-2 |
| 3.2.1 Maximum likelihood estimation | 3-3 |
| 3.3 Methodology for the present study | 3-6 |
| 3.3.1 Selection of the training area | 3-6 |
| 3.3.2 Bayesian maximum likelihood classification of test samples | 3-9 |
| 3.3.3 Bayesian maximum likelihood classification of the Singrauli Image. | 3-9 |
| 3.4 Results | 3-10 |
| 3.4.1 Level - I classification | 3-10 |
| 3.4.2 Level - II classification | 3-11 |

| | |
|---|------------|
| 4. Neural Network Pattern recognition | 4-1 |
| 4.0 General | 4-1 |
| 4.1 Artificial Neural Networks | 4-2 |
| 4.1.1 Structural levels of Organization of Brain | 4-5 |
| 4.1.2 Models of a Neuron | 4-7 |
| 4.1.3 Types of Activation function | 4-8 |
| 4.1.4 Learning process | 4-9 |
| 4.1.5 The Perceptron Model | 4-11 |
| 4.1.6 Backpropagation Algorithm | 4-16 |
| 4.1.6.1 General | 4-16 |
| 4.1.6.2 Training | 4-19 |
| 4.1.6.3 Forward pass | 4-20 |
| 4.1.6.4 Reverse pass | 4-20 |
| 4.1.7 Kohonen's Self Organizing feature Maps | 4-20 |
| 4.2 Methodology for classification | 4-25 |
| 4.2.1 General | 4-25 |
| 4.2.2 Classification using Backpropagation learning | 4-25 |
| 4.2.3 Classification using Kohonen's self organizing maps | 4-29 |
| 4.3 Methodology for river extraction | 4-30 |
| 4.4 Results | 4-30 |
| 4.4.1 Backpropagation learning | 4-30 |
| 4.4.1.1 level -I classification | 4-30 |
| 4.4.1.2 Level -II classification | 4-34 |
| 4.4.2 Kohonen's algorithm | 4-38 |
| 4.4.3 River extraction using Backpropagation learning | 4-40 |

| | |
|--|------------|
| 5. A Comparative Appraisal | 5-1 |
| 5.0 General | 5-1 |
| 5.1 Classification Accuracy | 5-1 |
| 5.2 Computational time | 5-3 |
| | |
| 6. Conclusions and Future Recommendations | 6-1 |
| 6.1 Conclusions | 6-1 |
| 6.2 Recommendations for Future work | 6-3 |
| References | |

LIST OF TABLES

| Table Number | Title | Page Number |
|--------------|---|-------------|
| 1.1 | Geological sequence of the Singrauli basin (after Joshi and Pant, 1971) | 1-7 |
| 1.2 | IRS spectral bands and principal applications (Dept. of Space, India) | 1-9 |
| 3.1 | Means for level - I classification | 3-7 |
| 3.2 | Means for level - II classification | 3-7 |
| 3.3 | Standard Deviation for level - I classification | 3-8 |
| 3.4 | Standard Deviation for level - II classification | 3-8 |
| 3.5 | Confusion matrix for maximum likelihood classification Level - I | 3-13 |
| 3.6 | Confusion matrix for maximum likelihood classification Level - II | 3-13 |
| 4.1 | Confusion matrix for 500 iterations in one hidden layer | 4-35 |
| 4.2 | Confusion matrix for 1000 iterations in one hidden layer | 4-35 |
| 4.3 | Confusion matrix for 5000 iterations in one hidden layer | 4-35 |
| 4.4 | Confusion matrix for 10000 iterations in one hidden layer | 4-35 |
| 4.5 | Confusion matrix for 15000 iterations in one hidden layer | 4-36 |

| | | |
|-----|--|------|
| 4.6 | Confusion matrix for 20000 iterations in one hidden layer | 4-36 |
| 4.7 | Confusion matrix for level - II classification by Backpropagation algorithm | 4-36 |
| 4.8 | Confusion matrix of Kohonen's output | 4-39 |
| 5.1 | Accuracy of three classifiers for level - I classification | 5-3 |

LIST OF FIGURES

| Figure Number | Title | Page Number |
|---------------|--|-------------|
| 1.1 | Location of major coalfields of India | 1-5 |
| 1.2 | Singrauli area (FCC) | 1-8 |
| 1.3 | Semi urban environment of Kanpur (FCC) | 1-8 |
| 3.1 | Schematic diagram for Bayesian maximum likelihood classification | 3-5 |
| 3.2 | Level - I classified image by maximum likelihood classification | 3-14 |
| 3.3 | Level - II classified image by maximum likelihood classification | 3-14 |
| 4.1 | The pyramidal cell | 4-4 |
| 4.2 | Block diagram representation of nervous system | 4-4 |
| 4.3 | Structural organization of levels in the brain | 4-6 |
| 4.4 | Nonlinear model of neuron | 4-6 |
| 4.5(a) | Threshold function | 4-10 |
| 4.5(b) | Piecewise linear function | 4-10 |
| 4.5(c) | Sigmoid function | 4-10 |
| 4.6 | Single layer perceptron | 4-12 |
| 4.7 | Two overlapping, 1 dimensional Gaussian distributions | 4-12 |

| | | |
|------|---|------|
| 4.8 | Architetural graph of a multilayer perceptron with two hidden layers | 4-17 |
| 4.9 | Perceptron neuron | 4-17 |
| 4.10 | Two dimensional array of output nodes used to form feature maps | 4-21 |
| 4.11 | Topological neighborhood representation in Kohonen's algorithm | 4-24 |
| 4.12 | Schematic diagram of Backpropagation Training procedure | 4-31 |
| 4.13 | Accuracy Vs Number of iterations | 4-33 |
| 4.14 | Level - I classified image by Backpropagation algorithm | 4-37 |
| 4.15 | Level - II classified image by Backpropagation algorithm | 4-37 |
| 4.16 | Classified image by Kohonen's algorithm | 4-39 |
| 4.17 | River extracted using Backpropagation learning | 4-40 |
| 5.1 | CPU time vs Number of iterations | 5-4 |

Abstract

The present work is undertaken to investigate the potential of Artificial Neural Networks (ANN) in extracting the feature of interest for digital satellite imagery of Earth. In this context the unsupervised and supervised learning based neural networks are being attempted. Various aspects of the Backpropagation and the Kohonen's Self Organizing Feature Maps algorithms (SOFM) have been investigated. The results obtained are analyzed and compared in light with the results rendered by conventional Maximum Likelihood Classifier.

The IRS -1B data in four spectral bands belonging to Singrauli Coal basin and semi urban environment of Kanpur is adopted. The Singrauli Coal basin offers well discernible level - I and Level -II classes, which have been utilized to test the classification accuracy of ANN. The other image contains major north Indian river Ganges.

Backpropagation algorithm has offered best results after choosing optimum number of iterations for learning and presented overall classification accuracy of 95.07 %. Maximum Likelihood Classifier rendered better result than unsupervised Kohonen's SOFM. It was observed that smaller network topologies were better than the larger ones. River Ganges along with other surface drainage features has been well extracted using Backpropagation algorithm.

The study has offered encouraging results and established that Backpropagation algorithm has potential to be adopted for feature extraction from digital satellite imagery.

Introduction

1.1 General

Pattern recognition may be characterized as an *information reduction, information mapping, or information labeling* process. Most of the land cover features found in our ambience are in form of some pattern. Thus, there is little to prove in the fact that pattern recognition is actually based on identification of patterns. Further to extract a pattern one may use some sort of measurement or analysis. They may be symbolic, numerical or both. *Feature Selection* is the process of choosing input to the Pattern recognition (PR) system and involves judgment. The objective of *pattern recognition* and *classification* is to distinguish between different types of patterns. Much of the concept of PR is based on the concept of similarity. *Classification* essentially is to assign input data into one or more of prespecified classes based on extraction of significant features or attributes and the processing or analysis of these attributes. *Recognition* is the ability to classify. *Description* is the alternative to classification where a structural description of the input pattern is desired. It is common to resort to linguistic or structural models for description. A *pattern Class* is a set of patterns originating from the same source. *Preprocessing* is the filtering or transforming of the raw data to aid computational feasibility and feature extraction and minimize noise.

In recent past, the Artificial Neural Networks (ANN) has been recognized as an efficient tool for decision making. The ANN model consists of a *variable interconnection of simple elements, or units*. Training is stored in form of network interconnections. It is expected that once trained the ANN would be able to predict correct 'associative' behavior, when presented with new patterns to recognize or classify. These systems are dynamic systems whose interconnection values change with time. Learning in Neural Networks may be supervised or unsupervised. Several neural network structures are useful for a class of PR

problems such as *feedforward network* which is used for supervised learning. Hopfield network is also used for supervised learning but is a recurrent network. Some of the unsupervised ANNs are *Kohonen's self organizing feature maps* (KSOFM) and *Adaptive resonance theory* (ART).

Computerized information extraction from remotely sensed imagery has been applied successfully over the last two decades. The data used in the processing have mostly been multispectral data, and the statistical pattern recognition methods (multivariate classification) have been extensively used. Over the last decade, advances in space and computer technologies have made it possible to amass large amounts of data about the Earth and its environment. The satellite data is available from multi sources, multialtitude, multiband and multiresolution of the same scene. These are collectively called multisource data.

The primary object of using all these data is to extract more information and achieve higher accuracy in classification. However, on close examination of the conventional multivariate classification it is clear that these methods cannot be satisfactorily used in processing multisource data. This is due to several reasons. One is that the multisource data cannot be modeled by a convenient multivariate statistical model since the data is of multitype. They can, for example, be spectral data, elevation ranges, and even nonnumerical data such as ground cover classes or soil types. The data are not necessarily in common units, and therefore scaling problems may arise. Another problem with statistical classification methods is that the data sources may not be equally reliable. This means that the sources need to be weighted according to their reliability, but most statistical classification methods do not have such mechanism. This all implies that methods other than conventional multivariate classification must be used to classify multisource data.

Various heuristics and problem-specific methods have been proposed to classify multisource data. However, present study is dedicated to developing more general methods which can be applied to classify any type of remotely sensed data. In this respect, two approaches will be considered; a statistical approach (parametric) and a neural network approach (distribution free).

The neural networks, or connectionist, approach was first introduced as a theoretical method of *Artificial Intelligence* (AI) in 1960s. However, limitations in simple systems were recognized by Minsky and Papert (1969) and the concept gave way to the symbol system approach for the next two decades. The idea has recently been revived due to advances in hardware technology allowing the simulation of neural networks and the development of nonlinear multilayered architectures (Rumelhart et al., 1986). In a remotely sensed imagery various features may be distinguished by gray level differences or using variation in the texture. In the present work, the entire computation involves working with Earth cover reflectance, which are gray level values received from the satellite. This is the simplest way to characterize the variability in an image segment. This gray level set becomes the feature set on which classification is done. An important advantage of this approach is the speed with which images can be processed since feature identification is computation independent. The main disadvantage is the size of the feature set. This method is only feasible for small image segments since the computational intensity increases massively for larger images. This also becomes a lacunae while using neural networks. It is seen that the size of networks should be kept as small as possible because larger networks involve more computation and also show poorer performance while converging (Lippman, 1987).

It has been felt by a large group of researchers in the remote sensing community that neural networks is a useful tool for image classification. The biggest drawback in this method is the large training time requirement for minimizing *square mean error*. Implementation of the training and classification algorithms on a massively parallel computing system would greatly enhance the applicability of the method. As computer hardware become more advanced and powerful computationally intensive applications such as neural networks become more attractive.

The present study is dedicated to explore the possibility of using ANN's speed coupled with its flexible decision capability based on smaller training set in context to IRS-1B digital image and analyze the results in light with outcome of Bayesian's maximum likelihood classifier.

1.2 Study Area

1.2.1 Singrauli coal basin

The area chosen to test present work is the Singrauli coal basin. The basin has an area of 2200 sq. km of which approximately 4 percent is located in Sonbhadra district of U.P. and the rest extending into Sidhi and Sahdol districts of M.P. (Fig 1.1). In India the thermal power generation has received an impetus in the recent times. In this context, Singrauli coalfield has gained prominence and presently occupies unique place in the coal map of India. It has a unique distinction of having the thickest coal seam of India, situated in Jingurdah mine with the thickness of 132 m (Tripathi, N.K., April 1994). The zone is having five Super Thermal Power Stations and series of industrial establishments around these thermal power stations. Thus, the region in the recent times has witnessed a rapid of economic development.

In the last three decades geological mapping has been in progress and extensive drilling has enabled the delineation of several coal seams including the coal seam in Jingurdah. The basin is divided into sub-basins, known as the Moher sub-basin in the East and the Main sub-basin in the West, the dividing line approximately conforming to the longitude $82^{\circ} 30' \text{ E}$. The coal field stands out as a prominent plateau in contrast to the adjacent plains. The NE part has been chosen as the study area. The sequence within the coal field in this region is essentially of Lower Gondwana (Upper Carboniferous - Permian) formations, An EW trending fault located in the northern part of the area limits the extension of these formations and marks the boundary with the Precambrian metamorphic rocks. The Precambrian formations are composed of gneisses, slates and phyllites. The generalized sequence (after Joshi and Pant, 1971) of the lower Gondwana formations in the study area is indicated in Table 1.1. The NE part of the basin is chosen as the study

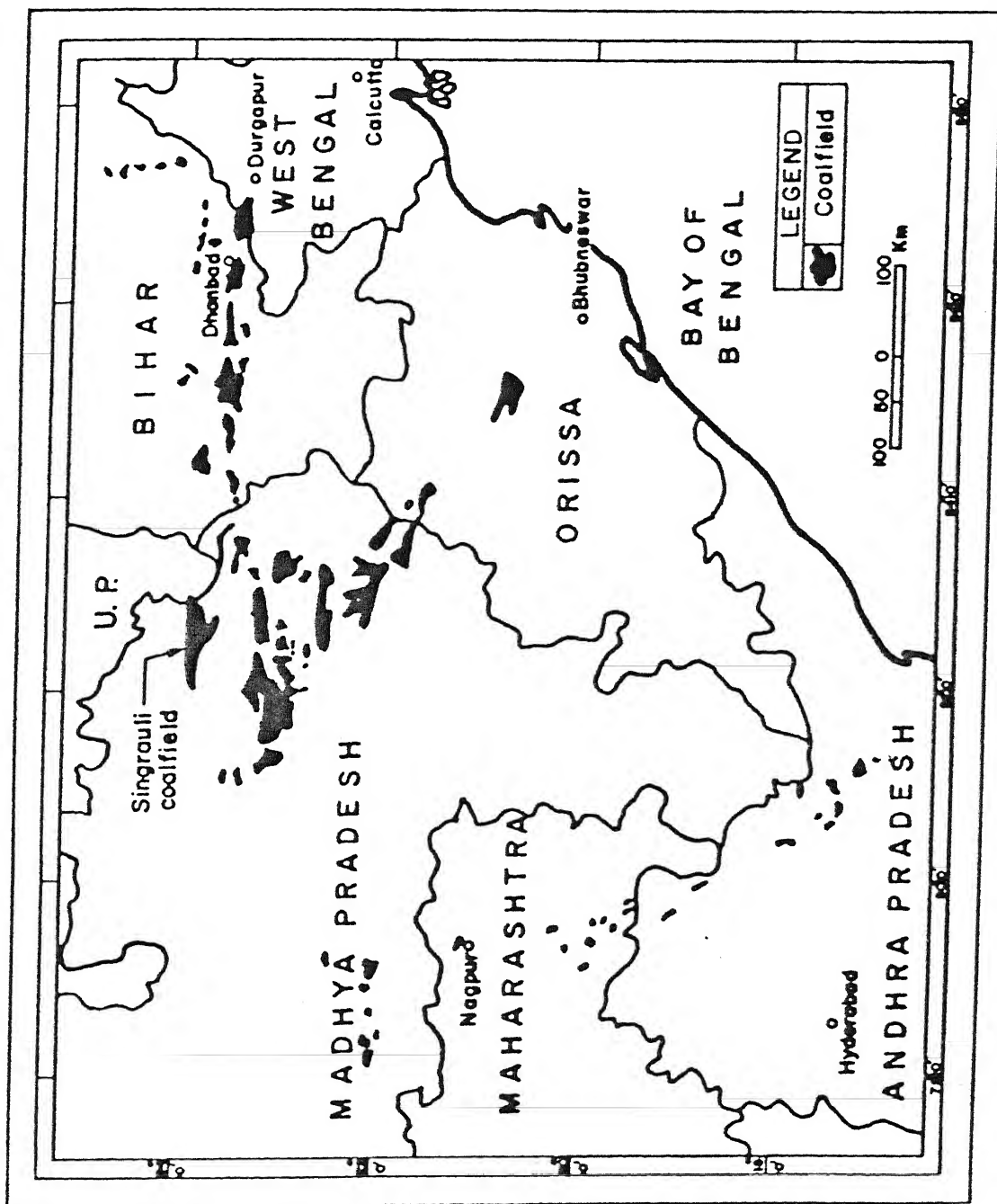


Fig.1.1: Location of major Coalfields of India (Source: CMPDI, 1984)

area. As indicated in the Table 1.1, continued subsidence of the sediments along the boundary fault appears to have increased the tilt which has resulted in these formations dipping towards the fault (Tripathi, N.K., 1994). Accumulation of greater thickness of sediments close to the boundary faults in the Gondwana basins has attributed to subsidence (Basu and Srivastava, 1980).

Talchir Formations

These are exposed mostly in the eastern and south-eastern part of the study area. The Talchir sequence has a basal diamictite horizon composed of striated pebbles of sandstone, jasper, gneisses and basic rocks (Murthy, 1957).

Barakar formations

The Barakar sequence includes medium to coarse grained sandstones (arkose type) interbedded with thin shales and clays. The sandstones are loosely cemented and often have ferruginous cement that lends a characteristic brown colour (Tripathi, N.K., 1994).

Barren measures

As the name indicates, these formations are devoid of coal. The sequence consists of coarse grained sandstones with pebbles, at places ferruginous in composition. In the study area, these formations are exposed around Jingurdah village, where the sequence has been reported to be 125 m thick (Raja Rao, 1983).

Raniganj formations

In the Jingurdah area, these formations have an arcuate disposition and attain a maximum thickness of 400 m as reported by earlier workers. The sequence essentially is of sandstones associated with clays and carbonaceous shale.

Major landuse features identified for the purpose of classification are coal mines, coal dumps, water (turbid water and clear water), vegetation (forest and sparse), rocks

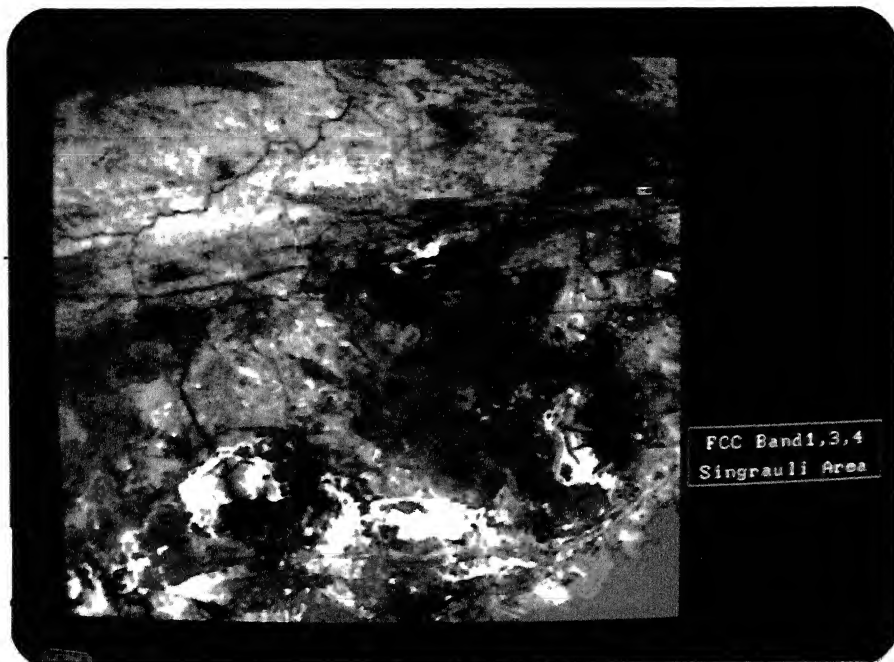
(quartzite and sandstone). These features are clearly identifiable in the false color composite (FCC) of the area shown in Fig 1.2. Bands 4, 3 and 1 have been used to generate the FCC.

1.2.2 Semi urban environment of Kanpur

The region has river Ganges which is a major river feature of the northern region of the country. Some other features such as sand bars, flood plains and agricultural fields are also present in this area with semi urban environment in the vicinity. The feature of interest was river Ganges for our study hence a 262×262 window from the image has been extracted for this test study. The FCC of this area is shown in Fig. 1.3.

TABLE 1.1 : Geological sequence of the Singrauli basin (after Joshi and Pant, 1971)

| Age | Formation | Lithology | Thickness (m) |
|---------------------|-----------------|--|------------------|
| Permian | Raniganj | Sandstone, Shale and Coal Seams | 380 |
| | Barren Measures | Coarse Sandstone | 8-12 |
| | | Purewa Top Seam Sandstone | 0-60 |
| | Barakar | Purewa Bottom Seam Sandstone and Shale | 10-14 45-75 |
| Upper Carboniferous | Talchir | Kota Seams Sandstone and Shale | 150-250 |
| | | Sandstone and Shale | not proved |



**Fig 1.2 Singrauli Area
(FCC 4, 3, 1 of IRS - 1B LISS II)**



**Fig 1.3 Semi urban environment of Kanpur
(FCC 4, 3, 1 of IRS - 1B LISS II)**

CENTRAL LIBRARY
I. I. T., KANPUR
No. A. 2262

1.3 Data Acquisition

1.3.1 Singrauli area

In the present work, data from IRS - 1B has been used. Indian Remote Sensing Satellite (IRS - 1B) was launched on August 29, 1991 in a Sun synchronous orbit at an altitude of 904 kms. The satellite carried three sensors (one pushbroom camera (LISS - I) of 72.5 m resolution and two cameras (LISS - II) of 36.25 m resolution), images obtained belong to four spectral bands in the visible and near IR region. The spectral characteristics of IRS - 1B are presented in the Table 1.2. The study area is covered on IRS - 1B index map by path number-row number 23-51 and the data pertains to April 25, 1992.

The study region lies in the map, issued by the Survey of India, numbered 63 L on a scale of 1:250,000.

1.3.2 Semi urban environment of Kanpur

For extraction of the river, the Kanpur region lying between ($80^{\circ}15'E$, $26^{\circ}37'30''$) and ($80^{\circ}22'30''E$, $26^{\circ}30'$), is selected which is covered in the Survey of India toposheet no. 63 B/6/SW. A 512×512 image of IRS - 1B has been selected for study.

Table 1.2: IRS spectral bands and principal applications (Dept. of Space, India, 1989)

| BAND NO. | SPECTRAL RANGE (μm) | SPECTRAL LOCATION | PRINCIPAL APPLICATIONS |
|----------|-------------------------------------|----------------------|---|
| 1 | 0.45-0.52 | BLUE | Sensitivity to sedimentation, deciduous/coniferous forest cover discrimination. |
| 2 | 0.52-0.59 | GREEN | Green reflectance of healthy vegetation. |
| 3 | 0.62-0.68 | RED | Sensitivity to chlorophyll absorption by vegetation, differentiation of soil and geological boundary. |
| 4 | 0.77-0.86 | NEAR- INFRA RED | Sensitivity to green biomass and moisture in vegetation, land-water contrast. |

1.4 Computing Platforms for the Study

1.4.1 Hardware used

The softwares have been developed on the HP - 9000/735 series, locally known as "Agni" and for those involving graphics, the HP -9000/330 series workstations have been used. Further, work has also been carried out on the PC machines. The standard hardware configuration of the system includes PC486 DX2 66mhz, SVGA COLOR monitor, monochrome monitor, 270 MB HDD and 8MB RAM.

1.4.2 Software Details

The entire programming for the work has been done in C by the author. For displaying the images, Starbase graphics has been used. ILWIS (Integrated Land and Water Information System) version 1.41 was also used in the present work. ILWIS is a GIS (Geographic Information System) that integrates image processing and spatial analysis capabilities, tabular databases and conventional GIS characteristics. The text for this work has been written on the MS-WORD and the graphs have been drawn using MS- EXCEL.

1.5 Scope of the Present Work

The main objective of the present study is to investigate the usefulness of ANN in recognizing the various features in digital remotely sensed images. This offers an intensive scope of investigating supervised and unsupervised neural network and their characteristics influencing the computational complexity and the accuracy offered.

To test the efficiency of ANN over their counter parts in Statistical Pattern Recognition, a common data set comprising of IRS-1B image in 4 bands (0.52 - 1.1 μm) of Singrauli has been adopted.

The outcome of study is likely to crystallize the suitability of pattern recognition approach of ANN for multiband remotely sensed data.

1.6 Organization of the Work

The material contained in the thesis is organized and presented in six chapters. Chapter one presents the introduction. Details of the Singrauli coal basin, introductory remarks on ANN and general description, data used and scope of study is also discussed . In Chapter two, a brief literature survey of the earlier work using neural networks in remote sensing and also the conventional classifiers, is presented.

Theoretical concepts of statistical pattern recognition are discussed in Chapter three. This chapter also contains the methodology followed in this work and results obtained.

Chapter four deals with Artificial Neural Networks and their application to remote sensing. The theoretical concepts of ANN, methodology for the current work and results obtained are presented.

Chapter five contains a comparative appraisal of the traditional classifiers and ANN classification over a common data set.

The findings of the work are summarized in Chapter six with recommendations for future work.

Literature Review

2.1 Image Classification

The overall objective of image classification procedure is to automatically categorise all pixels in an image into land cover classes or themes (Lillesand and Kiefer, 1994). For remote sensing purposes, normally the multispectral data available from the satellite is used, and the spectral pattern within the data for each pixel is used as the numerical basis for categorisation. This is realisable simply because different feature types have different digital number (DN) values depending on their inherent spectral emittance and reflectance properties. Thus, a spectral pattern is not at all geometric in nature. The procedure that utilises pixel-by-pixel spectral information as the basis for automated land cover classification is called *Spectral Pattern Recognition* (Lillesand and Kiefer, 1994).

Spatial Pattern Recognition involves the categorisation of image pixels on the basis of their spatial relationship with pixels surrounding them. Spatial classifiers find their basis in aspects such as image texture, pixel proximity, feature size, shape, directionality, repetition, and context. This kind of analysis involves complex mathematical jargons requiring intensive computation (Venkateshwarlu, 1988).

Temporal pattern Recognition uses *time* as an aid in feature identification. This technique is in fact, extremely helpful in agricultural crop surveys, flood and forest mapping. In this work the *spectral classification* strategies are being investigated. The traditional methods of classification mainly fall into two categories, *unsupervised* and *supervised*. The *unsupervised* approach attempts to identify spectrally homogeneous clusters of pixels in the image. These spectral groups are not having any labels hence the user might have to undergo another exercise to associate them into known features by using some other source of information. This approach is often referred to as clustering. In the *supervised*

approach, the image analyst supplies his algorithm with the knowledge of the various classes present in the scene, by quantifying them. To accomplish this, the analyst first selects the training areas from the image that represent the various classes present which act as numerical interpretation keys and help in assigning unknown pixels to the class they most likely belong (Tripathi, N. K., 1987).

2.1.1 Supervised classification

As discussed above in this classification the user “supervises” the process. This is done by using *training data* as input. In this the various classes are given with the respective pixels as input. Great care is to be taken in selecting the training data as some studies suggest that the major factors affecting the classification accuracy is the quality of the training data. The data used for training must not include other classes, yet must include a representative spread of pixels from the class. Thus, we may summarise the three basic steps involved in a typical supervised classification procedure. In the *training stage* the analyst identifies representative training areas and develops a numerical description of the spectral attributes of each land cover type of interest in the scene. Next stage is the *Classification stage* where each pixel in the image data set is categorised into the land cover class it most closely resembles. The category label assigned to each pixel in this process is then recorded in a interpreted data set (an “output imagery”). Once the entire data is categorised, the results are presented in the *output stage*. The results obtained are digital in nature and may be presented in a number of ways. Three typical forms of output products are thematic maps, tables of full scene and subscene area statistics pertaining to various land cover types, and digital data files that can be included in the GIS (Lillesand and Kiefer, 1994).

Various classification procedures used for spectral pattern recognition are *Minimum-Distance-to-means Classifier*, *Parallelepiped Classifier*, *Gaussian Maximum Likelihood Classifier*. These are all statistical classifiers. Gaussian maximum likelihood classifier has established itself to be one of the most accurate classifier (Foody, 1995). For present work this classifier is selected and is discussed in sufficient detail in Chapter 3.

2.1.2 Unsupervised classification

As discussed earlier, unsupervised classifiers do not utilise training data as the basis for classification. Instead, this family of classifiers involves algorithms that examine the unknown pixels in an image and aggregate them into a number of classes based on the natural groupings or clusters present in the image values. The basic premise is that values within a given cover type should come together in the measurement space, whereas the data in different classes should be comparatively well separated. These classes are essentially *spectral classes*, the identity of which is not known initially. Thus, the classified data has to be compared with some other source like a toposheet and then assign the respective clusters with known labels. There are numerous clustering algorithms available such as *K-means*, *ISODATA clustering* (Lillesand and Kiefer, 1994).

2.2 Overview of the Earlier Works

A wide range of digital classifiers are used to classify remotely sensed satellite imagery. The result of classification, however, may vary from classifier to classifier depending on their efficiency. But that is not all, classified output is often a function of various other factors, viz., nature of the remotely sensed data and the type of training set selected. These issues are important because most of the statistical classifiers are bound by some assumptions that limit their efficiency. Say, a Maximum Likelihood classifier (MLC) is based on the normal distribution of the data. Pragmatically, this assumption may not be valid in many cases as the data may follow some different distribution function and correction of normality is impossible. Even if the normality condition is valid, the performance is dependent on the nature of the training data needed, e.g., it is advisable to use a training data, thirty times of the number of features in an image (Mather 1987, Piper 1992). Such large requirement may not be satisfied if there is a scarcity of samples. Furthermore, presence of noise or missing of data may give unsatisfactory results. Since, the data we receive from the satellite is multispectral in nature, the multivariate classification is used. There is, however, computational difficulty of using a convenient statistical model due to multispectral nature of the data. The data is not in common units hence scaling problems may arise. Further, reliability of the data source is not guaranteed.

In such a case the data sources need be weighted equally. But most of the statistical approaches do not consider this aspect.

Indeed, alternative classification approaches are now being looked for. A large number of approaches have attracted attention of remote sensing community including those based on fuzzy sets (Kent and Mardia 1988, Wang 1990) and evidential reasoning (Moon 1993, Peddle 1993) but one which is gaining popularity is Artificial Neural Networks (Foody, G. M., 1994).

The most apparent advantage of ANN is that it is more robust than most of the statistical classifiers (Hepner et al., 1990). They are also more tolerant of the missing data and noise, and can adapt over time. They do need considerable amount of time in training. However, once trained, they may be more efficient computationally than the conventional classifiers, especially if run on *parallel distributed computing system*. There are several other projected advantages of neural nets (Lippmann, 1987; Hush and Horne, 1993; Haykin et al. 1991;):

- An intrinsic ability to generalise;
- Ability form highly non-linear decision boundaries in the feature space and therefore has the potential of outperforming conventional methods.

In the traditional classifiers, MLC is the most widely used classifier. The popularity of the MLC is due to a number of characteristics (Swain and Davis, 1978; Schowengredt, 1983; Richards, 1986). First, the maximum likelihood decision rule is intuitively appealing because the most likely outcome among candidate outcomes is chosen. Second, the decision rule has a well-developed theoretical foundation, and for normally distributed data is mathematically tractable and by many measures statistically desirable. Third, a *maximum likelihood classification* can readily accomodate covarying data, a common occurrence with satellite image data. Finally, MLC have been proven to perform well over a range of cover types, conditions, and satellite systems (Swain and Davis, 1978; Richards, 1986; Lillesand and Kiefer, 1987).

Despite the advantages of MLC, most implementations have exhibited at least one serious drawback, namely, long classification times. For N spectral bands and T training sets, computing the maximum likelihood for each N -tuple (pixel measurement vector) of image data requires at least $[(N^2 + N) * T]$ multiplications and $[(N - 1) * (N + 1) + 2*N]$ additions in the most commonly implemented form of the maximum likelihood decision rule (Richards, 1986). Accordingly, per-pixel maximum likelihood classification requires billions of calculations when applied to large-area high resolution satellite image data.

Many researchers have pursued studies to improve speed of working of the MLC. There are several hardware approaches. For instance, increased processor clock speed, using enhanced numeric co-processors and a third option is adaption of parallel processing technology. Apart from these "hardware" approaches, some analysts have proposed simple algorithms to increase MLC's speed while retaining its advantages. One such work was done by Lillesand and Bolstad (Jan, 1991). They suggested an improved table look-up technique for maximum likelihood classification on large images. Their method was powerful simple, portable and could be run in limited memory desktop computer environment. This reduced the time of implementation 20-fold.

Hepner et al. (1990), has examined the potential for the application of neural network to satellite image processing. The study was also performed with another objective of providing a preliminary comparison of training site data inputs and generalised land cover results for conventional supervised classification and ANN classification. The results of the study indicate that the ANN can classify imagery better than a conventional supervised classification using identical training sites. A single training site per class ANN classification was found to be comparable to four training site per class for conventional classifier. The conventional supervised classification using the single minimal training site was very inferior to the ANN classification. They clearly demonstrate that ANN is potentially more robust than the conventional classifiers. However, ANN was found to be computationally more intensive and the real time results can be obtained only from advanced hardware implementation or fully parallel processing software environment. In this study Thematic mapped imagery consisting of the visible spectral bands 1, 2 and 3 and the near-IR band 4 have been used. The network used consisted of a 3 by 3 by 4 array of neurons as the input, ten neurons in the hidden layer and four output neurons. A minimal

neurons as the input, ten neurons in the hidden layer and four output neurons. A minimal training set was chosen for both the classifications. This was a 10-pixel by 10-pixel site for each of the four classes. It took 3.1hrs CPU time to train the network and 15 mins to test, while it took 60 mins to train and classify using the MLC.

Foody G.M. et al. (April, 1995) have used *feed forward* artificial neural network using a variant of *back propagation* algorithm was used to classify agricultural crops from synthetic radiator radar data. The performance of the classification data was assessed with respect to a conventional statistical classifier, discriminant analysis. He concluded that ANN consistently provided a higher classification accuracy than did the discriminant analysis, indicating that it is more accurate to characterise class appearance. The differences, however, were only significant if the data was non-normally distributed. Further if *apriori* information was made available to the discriminant analysis then the classification accuracy of the ANN was not significantly different. Non-representative training data leads, as expected, to significant differences between training and testing classification accuracies, and the effect was fairly similar for both the ANN and Discriminant Analysis (DA). For the classification of the Synthetic Aperture Radar (SAR) data, a three layer artificial neural network architecture was used, comprising four input units, three hidden units, seven output units, and a bias unit. The classification was done under the assumption that each class had equal a priori probability of occurrence. To investigate the effect of unclassified classes on the training of the network, a separate methodology was followed. About 90 cases were taken to train the network, from the three most dominant classes. For testing 68 cases from all seven classes were taken. It was found that the neural network could train with greater accuracy though the discriminating analysis classified with marginally better accuracy. The training accuracy achieved was 94.44 percent for ANN and 98.89 percent for DA. The testing accuracy achieved was 47.06 percent for ANN and 44.12 percent for DA.

The quality of description of class appearance generated in the training stage is dependant on representativeness and statistical distribution of the data. A simulated data set was used to assess the effect of these factors. This comprised of four classes with each having 100 elements. Each class was generated with a normal distribution. The data was split into 33 and 67 cases, in each class such that the former was used for training. A 1-3-4 network

was used. In the first analysis training data of each set was sampled such that they were representative of each class from which they were drawn. Representativeness was tested by Mann whitney U test. Then non-representative samples were chosen to train . Finally, normally and representative, normally and non-representative, non normally and representative, and non normally and non-representative were selected. The results were as follows:

| Normally and representative | | |
|-------------------------------------|--------------|------------|
| | Training (%) | Testing(%) |
| ANN | 78.79 | 78.36 |
| DA | 76.52 | 74.63 |
| Normally and non representative | | |
| ANN | 81.82 | 67.16 |
| DA | 80.30 | 67.54 |
| Non normally and representative | | |
| ANN | 60.61 | 62.31 |
| DA | 55.30 | 55.97 |
| Non normally and non representative | | |
| ANN | 65.15 | 55.97 |
| DA | 51.36 | 49.25 |

Tzeng et al. (1993), have used a slight variant of feed forward network called *co-operative learning neural network*. This network consists of a category extraction network and a unification network. The Back propagation algorithm is employed for the learning of the network. Number of extraction networks depends upon the number of output categories defined later. The extraction network is selected according to the category. In the analysis to compare the learning convergence and accuracy, the hidden units are 20 and the input units per spectral band are 9 for 9 neighbouring pixels. They found that the proposed network was effective in accelerating the learning convergence. The MSE (mean square error) of the proposed network is smaller than the conventional one as the input patterns of the unification network are already normalised in the previous extraction networks. Thus complex non-linear networks can be recognised by passing through the extraction

networks. Thus, they concluded that the proposed network could recognise better than the conventional three layered network. Further they discovered that, since, it was possible to add more information to these networks like segment information, the accuracy improves marginally (51.7 percent compared to 48.9 percent when only spectral band information was fed in the network).

Foody G.M. (1995) stressed the importance of prior class in increasing of the accuracy of classification. He concluded that often small data set is desirable since large data sets are not available, but the performance of the conventional classifiers do not function well when supplied with small data set. In such cases the use of ANN is indispensable as they are shown to classify the data more accurately. Clearly, this accuracy would still be lower than the larger training set accuracy. Hence, the a priori probabilities were incorporated. It was observed by him that the accuracy on the incorporation of a priori probabilities increased the accuracy from 27 percent to 58.4 percent. Also, it was learnt that using the a priori probabilities with the discriminant analysis but with a larger training set resulted in a similar accuracy. Essentially, he used the X-band HH-polarised SAR images for the area on four dates. After applying radiometric correction, 144 selected fields were selected for analysis. One field of each class was selected to form the training set. Rest of the fields were used for testing. The mean digital number values for each of the four fields were determined for all of the four images. They were scaled between 0.0 and 1.0. Then these were fed as the input in the neural network which comprised of 4 input neurons, 3 hidden layer neurons and 7 output neurons. Training was achieved by *Quickprop* learning algorithm, a variant of the widely used backpropagation algorithm (Fahlman, 1988a). The activation function used was a *sigmoid function*. The learning rate, decay and maximum growth were set to 0.5, 4 and 1.75. About 1000 iterations were performed. Once trained, the testing data was entered. Classes were allotted according to the highest activation level of the output nodes. A second set of class allocations was done by modulating the output with the a priori probabilities.

Benediktsson et al. (1993) have compared the neural network learning procedures and statistical classification empirically in classification of multisource remote sensing data. They have attempted to introduce reliability analysis in the traditional Bayesian theory approach. To increase the influence of the more reliable source they have introduced an

exponent term in the membership function. Reliability of the sources are quantified by appropriate reliability measures like separability of classes, classification accuracy of a data source and equivocation. Finally, these values of measures are associated with the reliability factors. They incorporated two neural network approaches, Delta rule and Generalised Delta rule. They obtained encouraging results from the neural network approach. Firstly, that Generalised Delta rule showed great potential as a pattern recognition method for multivariate sources. Further it was found to be distribution free. However it is computationally more complex. When the sample size is large it can take more time to learn. The overall accuracies obtained are listed in table:

| | Training (%) | Testing (%) |
|----------------------------|--------------|-------------|
| Maximum Likelihood | 60.9 | 49.2 |
| Minimum Euclidean distance | 58.2 | 46.6 |
| Mahalanobis | 60.8 | 49.7 |
| Artificial Neural Network | 95 | 52.5 |

Paola (1995) performed a detailed comparison of the backpropagation neural network and maximum likelihood classifiers for urban land use classification. The backpropagation routine used in this work had an adaptive learning rate and momentum. This was done with the aim to keep the learning rate at a level below the point at which it causes instability. After the user defined number of training cycles, the mean squared error is compared with that of previous cycle and if found more then the learning rate and the momentum factors are halved. For the input to the network, the pixel data was scaled from 0.0-1.0. For training, an output of 0.9 was used to represent the correct class while an output of 0.1 represented other classes. Initial weights were chosen randomly. Once these parameters were set, the number of hidden layers was calculated for which it was assumed that both the classifiers had the same number of parameters (degrees of freedom). This resulted in obtaining six neurons for each of the three layer network. In his study he confirmed the superiority of neural networks in the accuracy while having a drawback of consuming lot of time in training. The test site accuracy was predicted up to a maximum of 90 percent in the case of neural networks and about 89.5 percent in the case of maximum likelihood classification. A comparison of classification accuracy by using

various training sets varying in the number of training samples established that on increasing the number of training samples, accuracy of prediction increases

Statistical Pattern Recognition

3.1 Introduction

Machine intelligence has gained prominence in various aspects of technological development. Pattern recognition (PR) techniques are often an important component of *intelligent systems* and are used for both data processing and decision making. PR is not an approach, but rather a broad body of often loosely related knowledge and techniques. There are essentially three approaches to PR, namely, Statistical PR, Syntactic (structural) PR and ANN PR. Since, no single technology is an optimum solution for any PR problem hence in present work two approaches, Statistical PR and ANN PR are investigated. This current chapter deals with Statistical PR and its application.

The problem of classification is basically on of partitioning the feature space into regions, one region for each category. To attain accuracy in this partitioning, one would like to minimize the probability of error, or, if some errors are more costly than others, the average cost of errors (Duda and Hart, 1973). In this case, the problem of classification becomes a problem in statistical decision theory. Statistical PR techniques classify patterns (or entities) based on a set of extracted features and an underlying statistical (perhaps ad hoc) model for the generates of these patterns. It might be nice if all Pattern Recognition problems could be approached by using single straightforward procedure namely: (1) Determination of feature vector (2) Training the system; and (3) Classifying patterns (Skalkoff, 1992).

In the present study, Bayes Maximum Likelihood Classification theory has been used. The computer code for the same has been written by the author in C programming language on the HP-9000/735 series computing systems. The area of study lies in the Singrauli coal

basin. The IRS-1B data for the study area has been used in this work (details discussed in section 1.3).

3.2 Supervised Classification using Bayes Maximum Likelihood Classification

Bayesian Maximum Likelihood Classifier is a well-developed method originating from statistical decision theory that has been applied to problems of classifying data. Bayesian decision formula is

$$q(i/v) = \frac{q(v/i)q(i)}{q(v)} \quad (3.1)$$

where $q(i)$: Apriori probability associated with class i ,

$q(v/i)$: Probability that a pixel from class i has value v ,

$q(i/v)$: Probability that a pixel with value v has class I , and

$q(v)$: The sum of $q(v/i)$ over all I

Bayes decision rule states that: *Given a pixel with value v and, for each class i , the probability $q(i/v)$, that the pixel is from class i , then the best class to assign the pixel to is the class for which $q(i/v)$ is maximum* (Niblack, 1986).

To design an optimal classifier one should know the a priori probabilities and the class conditioned densities. However, in Pattern Recognition problems one rarely has an exact estimate of these values. One may, however, have a vague qualitative idea when he combines the data from the toposheet. The problem therefore is to find some way to use this information to design the classifier. In Remote Sensing applications the dimensionality of the problems is too large. Hence, one cannot simply use the samples to estimate the *apriori* probabilities and the probability density functions. Thus, to reduce the severity of the problems the solution is parameterized. That is, to say that, normal distribution of probability densities is assumed with mean μ_i and covariance matrix Σ_i for class i , although

exact values are not known. Thus, the problem now simplifies to estimating of parameters μ_i and Σ_i , through selection of an appropriate function. The most common choice of functional representation of $q(v/i)$ is as Gaussian distributions, in which for 1-D case the parameters to be estimated are mean and standard deviation of the distribution. The general form of one dimensional Gaussian with normalized area is:

$$\frac{1}{(2\pi)^{1/2} \sigma} e^{-\left[\frac{(x - m)^2}{\sigma^2}\right]} \quad (3.2)$$

In our case, this becomes

$$q(v / i) = \frac{1}{(2\pi)^{1/2} \sigma_i} e^{-\left[\frac{(x - m)^2}{2\sigma_i^2}\right]} \quad (3.3)$$

where μ_i and σ_i are mean and standard deviation of the i^{th} class. For n band problem, μ_i becomes an n -dimensional vector and σ_i becomes a $n \times n$ covariance matrix and the scaling factor becomes $(2\pi)^{1/2}$.

3.2.1 Maximum Likelihood Estimation (MLE)

Maximum Likelihood views the parameters as quantities having fixed unknown values. The best estimate is defined to be the one that maximizes the probability of obtaining samples actually observed. The mean and standard deviation for the corresponding sample or training data is taken as the same for the *image*. We now discuss the general principle behind MLE.

Suppose that we separate a set of samples according to class, so that we have c sets of samples H_1, H_2, \dots, H_c with the samples in H_j , having been drawn independently according to the probability law $q(v/j)$. It is assumed that this has a known parametric form, and is therefore uniquely determined by the value of a parameter vector θ_j . For our case $q(v/j) \sim N(\mu_j, \Sigma_j)$, where the components of θ_j include the components of both μ_j and Σ_j . Hence, the problem now is to use the information provided by the samples to obtain good estimates for the unknown parameter vectors $\theta_1, \theta_2, \dots, \theta_c$.

It is assumed that the parameters for different classes are functionally independent, hence each class can be dealt separately.

Now, suppose H contains n samples, $H = \{v_1, v_2, \dots, v_n\}$. Then, since, these samples were drawn independently,

$$q(H / \theta) = \prod_{k=1}^n q(x_k / \theta) \quad (3.4)$$

This is the likelihood function of θ and is also called as the likelihood of θ with respect to set of samples. The *maximum likelihood estimate* of θ is, by definition, that value $\hat{\theta}$ that maximizes $q(H/\theta)$. However, for computation purposes it is usually easier to work with the logarithm of likelihood than the likelihood itself. Clearly, this does not affect the result as the logarithm is monotonically increasing. Maximizing the function yields the following result:

$$g(v / i) = q(i) \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-1/2 (v-m_i)^T \Sigma_i^{-1} (v-m_i)} \quad (3.5)$$

Thus, as discussed earlier, when the training data yields the parameter values, then we use the above function is used to obtain the class values by assigning the class corresponding to maximum $p(i/v)$ value. The term

$$(v - m_i)^T \Sigma_i^{-1} (v - m_i)$$

in the exponent is weighted distance from v to m_i (weighted by Σ_i^{-1}) called the Mahalanobis distance; the term

$$\frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}}$$

is the normalization factor to give the Gaussian distribution a unit area/volume; and $q(i)$ is the a priori probability to scale the result (Niblack, 1986).

The simpler expression after taking logarithm is

$$g(i / v) = \log_e q(i) - 1 / 2 \log_e |\Sigma_i| - 1 / 2 (v - m_i)^T \Sigma_i^{-1} (v - m_i) \quad (3.6)$$

The value for g is calculated for each pixel. If there are c classes then c values of g are obtained for each pixel. The class having minimum g value for that pixel is assigned to the pixel.

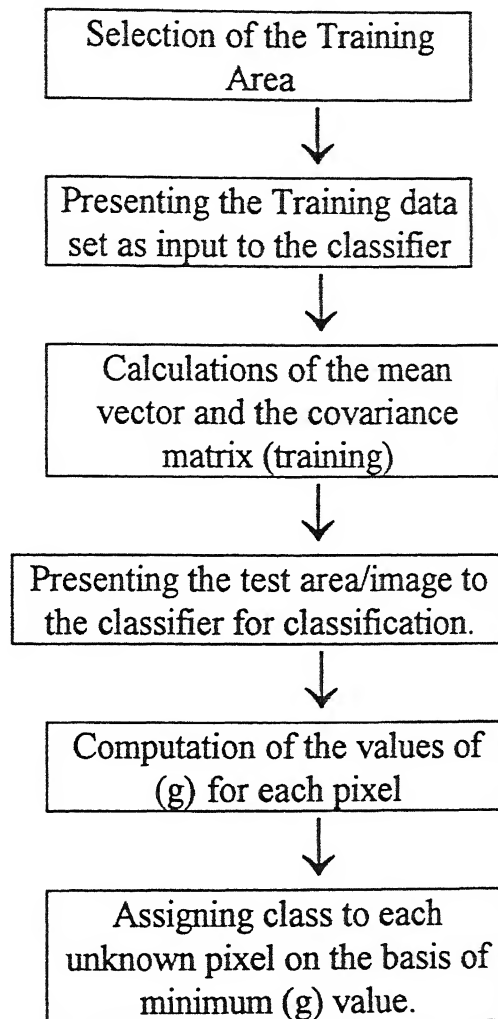


Fig 3.1 Schematic diagram for Bayes' Maximum Likelihood Classification

3.3 Methodology for the Present Study

The present study was conducted in three stages (Fig 3.1). Firstly, selection of the training area from the satellite imagery of the Singrauli region. This was first done for level - I classification and then later extended to level -II classification. This was followed by training of the classifier. Finally, a test area selected from the image was presented to the classifier to estimate the accuracy of the statistical classifier. The final image was also given as input to the classifier to obtain the classified image of the region.

3.3.1 Selection of the training area

The IRS-1B data was first displayed using a image display program. Various classes were identified visually using the toposheet of the area. Then, pixel values representative of various classes were noted down. The process was, actually, accelerated by coding a cursor aided program written in C and Starbase graphics. The program facilitates the image analyst to move the cursor on the image and select points of interest. The values corresponding to the selected points get stored in a file. This training area was then fed as input to the various algorithms as discussed later. The number of training samples for level - I classification was 120. There were six primary classes identified which are as follows:

- clear water
- turbid water
- vegetation
- coal mines
- coal dumps
- rocks

Table 3.1 Means for level - I classification

| | Band 1 | Band 2 | Band 3 | Band 4 |
|---------------------|---------------|---------------|---------------|---------------|
| Clear water | 88.4 | 64.45 | 82.1 | 37.5 |
| Turbid water | 104.65 | 73.05 | 95.40 | 47.25 |
| Rocks | 77.95 | 53.025 | 79.75 | 74.9 |
| Vegetation | 69.44 | 43.055 | 63.55 | 63.66 |
| Coal mine | 63.39 | 34.72 | 45.00 | 40.06 |
| Coal dumps | 108.50 | 76.90 | 119.69 | 102.5 |

Table 3.2 Means for level - II classification

| | Band 1 | Band 2 | Band 3 | Band 4 |
|--------------------------|---------------|---------------|---------------|---------------|
| Clear water | 88.4 | 64.45 | 82.1 | 37.5 |
| Turbid water | 104.65 | 73.05 | 95.40 | 47.25 |
| Sparse vegetation | 73.33 | 47.11 | 70.94 | 67.33 |
| Dense vegetation | 65.55 | 39.00 | 56.15 | 60.00 |
| Coal mine | 63.39 | 34.72 | 45.00 | 40.06 |
| Coal dumps | 108.50 | 76.90 | 119.69 | 102.5 |
| Quartzite rock | 78.65 | 55.15 | 85.45 | 74.90 |
| Sandstone | 77.25 | 50.9 | 74.05 | 71.40 |

Table 3.3 Standard Deviations for level - I classification

| | Band 1 | Band 2 | Band 3 | Band 4 |
|---------------------|---------------|---------------|---------------|---------------|
| Clear water | 1.273 | 1.234 | 2.268 | 2.0391 |
| Turbid water | 1.981 | 1.503 | 2.036 | 2.268 |
| Rocks | 1.893 | 1.895 | 3.136 | 2.463 |
| Vegetation | 4.193 | 2.621 | 3.238 | 2.972 |
| Coal mine | 2.09 | 1.742 | 2.99 | 4.62 |
| Coal dumps | 2.781 | 2.245 | 3.934 | 2.46 |

Table 3.4 Standard Deviations for level - II classification

| | Band 1 | Band 2 | Band 3 | Band 4 |
|--------------------------|---------------|---------------|---------------|---------------|
| Clear water | 1.273 | 1.234 | 2.268 | 2.0391 |
| Turbid water | 1.981 | 1.503 | 2.036 | 2.268 |
| Sparse vegetation | 5.71 | 3.411 | 3.795 | 3.29 |
| Dense vegetation | 1.605 | 1.45 | 2.56 | 2.616 |
| Coal mine | 2.09 | 1.742 | 2.99 | 4.62 |
| Coal dumps | 2.781 | 2.245 | 3.934 | 2.46 |
| Quartzite rock | 1.598 | 1.461 | 2.46 | 2.61 |
| Sandstone | 2.15 | 2.25 | 3.691 | 2.303 |

The number of training samples for level -II classification were 160. There were eight classes identified as follows:

- clear water
- turbid water

- dense vegetation
- sparse vegetation
- coal mines
- coal dumps
- quartzite
- sandstone

For the selection of the training area IRS band 3 was used. Tables 3.1 and 3.2 show the mean and Tables 3.3, 3.4 show the standard deviation statistics of level -I and level - II classifications respectively. Only those pixels lying in the range $[\mu - 2\sigma, \mu + 2\sigma]$ were selected as training area sample.

3.3.2 Bayesian maximum likelihood classification of test samples

The trained classifier is presented with test area having 300 samples in the level - I classification and 400 samples in the level - II classification. These do not have the classes assigned with them and the classifier predicts the class based on the criterion discussed in section 3.2.1.

3.3.3 Bayesian maximum likelihood classification of Singrauli image

Finally, the four band data, each having 512×512 pixels, is fed on the classifier. The classifier determines the identity of each pixel and assigns the appropriate class to it based on Bayesian maximum likelihood classification.

3.4 Results

3.4.1 Level - I classification

The results obtained from the classification are summarized in Table 3.5 in form of a confusion matrix. 'Confusion matrix' is a matrix, whose diagonal elements are number of samples correctly identified and elements in other columns are incorrect identification. The classifier determined classes for each of the 300 test samples. There were six classes identified for level - I classification. The overall accuracy of the classifier comes out to be 94.10 percent. The accuracy of clear water (class 1 in the Table 3.5) is 100 percent of classification. The accuracy of classification also depends on the type of training set taken (Lillesand and Kiefer, 1979). The training set of clear water could be selected with reasonable accuracy as it could be demarcated easily from the other classes. Further, its pixel values did not have any significant overlapping with other class. Besides clear water, another category of turbid water was identified. The turbid water (class 2 in the Table 3.5) is also present in the G.B. Pant reservoir which it also has extremely high temperature, rendering high reflectance values. This was also identified easily while selection of training data and the classifier yielded the test sample with 100 percent accuracy for test samples. The study region is characterized by patches of sparse vegetation and dense forests. For level - I classification, both types of vegetation are placed in one category (class 3 in Table 3.5). This category is labeled as vegetation. The training area was selected so that both the types got proper representation. The classification accuracy was 68 percent. It was observed that some of the pixels belonging to vegetation got classified in the class 'rocks'. It may be due to the reason that some training set elements of vegetation were collected from regions having sparse vegetation with possible rock exposures in training zone. Similarly, while selecting the test area a similar error is likely to have occurred. The rock features are classified with 96 percent accuracy with wrongly classified pixels lying in the clear water category. This might have happened due to some overlapping pixel values in rocks and clear water in band 1 and 3. The open cast mines (coal mine area, class 5 in Table 3.5) and the coal dumps (class 6 in Table 3.5) are, however, classified with 100 percent and 98 percent accuracy.

The classified image so obtained of the study area is shown in Fig 3.2. The six classes have come out reasonably well in the image, represented by different colours for each class. Clear water is shown as black colour in H-7. This is the G.B. Pant reservoir. The turbid water is represented blue colour, which comes as patch at G-6 lying just above the clear water body. The white patches scattered in garland shape around the mining areas of all over the image indicate the coal dumps. One such coal dump is in G-5. The coal mines are indicated by the dull white shades such as in C-5. The image is also characterized by many dark green patches which indicate vegetation. On such large patch, which is a forested land, lies between A-7 to C-5. A major part of the image has pink color which shows rocks. Some of the regions having vegetation (sparse) are also shown as rocks, for instance between E-2 and D-3, and between C-4 and C-6.

3.4.2 Level - II classification

Level - I broadly classified into six classes. However, these classes can be further divided into subclasses. For, instance, rocks has been subdivided into sandstone and quartzite, while vegetation into dense and sparse vegetation categories. This results in having total eight categories. The classifier was trained for a new training file consisting of 160 samples, as discussed earlier. The test file consisted of 400 samples. The classifier, yielded, to an accuracy of 94.25 percent. The clear water and turbid water were again classified to an accuracy of 100 percent. The sparse vegetation, however, is classified to an accuracy of 66 percent only. The pixels present in sparse vegetation were wrongly classified in three classes. These classes were dense vegetation (class 4 in Table 3.6), coal mines(class 5 in Table 3.6) and sandstone(class 7 in Table 3.6). The region is characterized by presence of coal mines near the forested land and also show low reflectance. The constraint in recognition of certain areas while selecting the training set is the main cause of misclassification and poor accuracy. Thus, besides natural limitation of the classifier's accuracy, this could be a probable explanation for mis-classification. If we examine the means of the sandstone and sparse vegetation (Table 3.2), they seem to be closely placed. This could have been the reason for the inaccurate classification. In sandstone (class 7) some pixels have been assigned to dense vegetation label and coal dumps, while having an

overall accuracy of 96 percent. Quartzite (class 8) has been classified fairly accurately upto 92 percent, while coal dumps (class 6) and coal mines (class 5) have attained 100 percent accuracy.

The classified image of the area is shown in Fig.3.3. The clear water is shown in blue in H-7 while the turbid water in black between H-7 and G-7. Coal dumps have come out in darker shade of blue with the largest patch lying in G-5. The dense vegetation has been shown in dark brown lying between A-6 to G-5. The exposed quartzite rock is present in the northern part of the study area shown by white colour. One such patch lies between B-2 and B-4. The reddish or brown patches indicate sparse vegetation. The light yellow patches lying between H-4 and G-6 represent sandstone. The green patches indicate the coal mines zones. Two large patches lie between C-4 to C-5 and D-7 to F-7.

Table 3.5 Confusion Matrix for Maximum Likelihood Level -I Classification

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 34 | 13 | 3 | 0 | 68 |
| 4 | 2 | 0 | 0 | 48 | 0 | 0 | 96 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy= 94.10 %

1- Clear water, 2-Turbid water, 3-Vegetation, 4-Rocks, 5-Coal mines, 6-Coal dumps

Table 3.6 Confusion Matrix for Maximum Likelihood Level - II Classification

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Accuracy (%) |
|---|----|----|----|----|----|----|----|----|---------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 33 | 4 | 3 | 0 | 10 | 0 | 66 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 100 |
| 6 | 0 | 0 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 7 | 0 | 0 | 0 | 1 | 0 | 1 | 48 | 0 | 96 |
| 8 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 46 | 92 |

Overall Accuracy= 94.25 %

1-Clear water, 2-Turbid water, 3-Sparse vegetation, 4-Dense vegetation
5-Coal mines, 6-Coal dumps, 7-Sandstone, 8-Quartzite

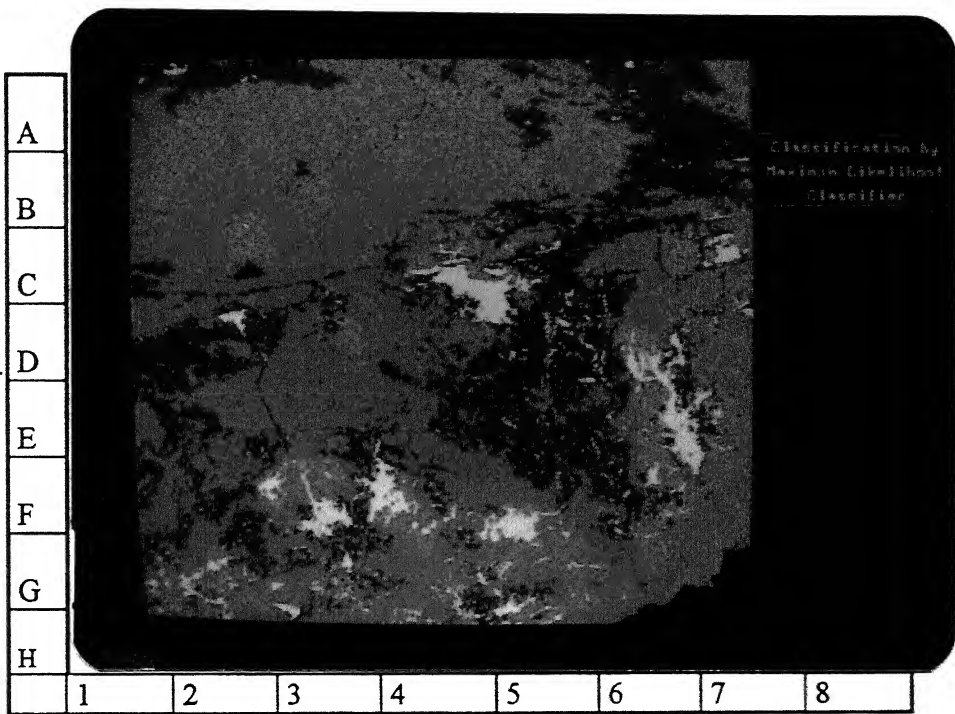


Fig 3.2 Level - I classified image by Maximum Likelihood classification

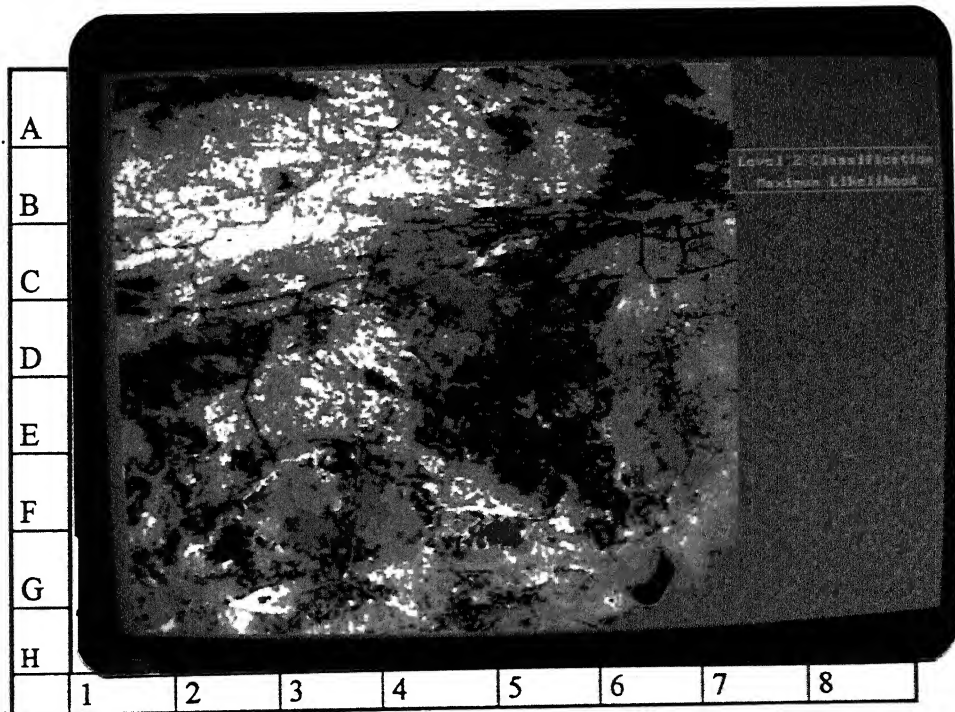


Fig 3.3 Level - II classified image by Maximum Likelihood classification

Pattern Recognition using Artificial Neural Networks

4.0 General

Historically, the two major approaches to pattern recognition are, the statistical (or decision theoretic) and the syntactic (or structural) approaches. The emerging technology of neural networks has provided the third approach (Schalkoff, 1992). Recently there has been a great resurgence of research in neural networks. New and improved neural network models have been proposed, models which can be successfully trained to classify complex data. In the remote sensing community, the question of how well the neural network models perform as classifiers, compared to statistical classifiers, is of considerable interest. The Remote Sensing Literature on Artificial Neural Network applications to *multispectral image classification* is relatively new dating back to about six years. The preliminary studies have established feasibility of the method (Benediktson et al., July 1990). Subsequent investigation were aimed at examining the classifier in more detail and compared to their results with the standard techniques such as maximum likelihood and k-means classifier. Some researchers found the traditional statistical classifiers superior (Benediktson et al., May, 1990), while a majority found that the neural network produced similar or superior classifications (Bischof et al., May, 1992). Neural networks have an advantage over the statistical classifiers that they are distribution free and no prior knowledge is needed about the statistical distributions of the classes in the data sources in order to apply these methods for classification. The neural networks also take care of determining how much weight each data source should have in the classification. A set of weights describe the neural network, and these weights are computed in an iterative training procedure. On the other hand, neural network models may be computation intensive, and need huge training samples to be applied successfully. In addition to this, their iterative training procedures usually are slow to converge. Also, neural network models have more difficulty than do statistical methods in classifying patterns which are not identical to one or more training patterns. The performance of neural networks in

classification is more dependent on having representative training samples, whereas the statistical approaches need to have an appropriate model for each class.

The present chapter first examines the theoretical concepts of neural networks followed by an overview of the methodology used in this study. Finally, there is an analysis and discussion of the results obtained in this study.

For this work, software for the algorithms used have been developed in C programming language by the author on the HP 9000/735 series computing systems.

4.1 Artificial Neural Networks

The fact that computation processes of the brain differ completely from the way a digital computer performs, motivated research in the Artificial Neural Networks more commonly known as *Neural Networks*. It was Ramon Y. Cajal who first introduced the idea of *neurons* in 1911 as the structural constituents of brain. The neuron, though, performs slower than the silicon chip i.e., an activity of silicon chip takes about 10^{-9} sec range and neuron events happen in 10^{-3} s range, however, brain makes up for this slow performance by having a colossal number of neurons and numerous interconnections in them: it is estimated that there must be of the order of 10 billion neurons in the human cortex, and 60 trillion synapses or connections (Shepherd and Koch, 1990). This results in an enormously efficient structure. Interestingly, the *energetic efficiency* of the brain is approximately 10^{-16} j (joules) per operation per second, whereas the corresponding value for the best computers in use today is about 10^{-6} joules per operation per second (Faggin, 1991).

The brain is a highly *complex, nonlinear, and parallel computer* (information processing system). It has the capability of organizing neurons so as to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence today. At birth, a brain has great structure and the ability to build up its own rules through "experience". The brain develops the most during the first two years of life while the experience keeps on compiling for years as the

development proceeds. During the early stage of development, about 1 million synapses are formed (Haykin, 1994).

Synapses are elementary structural and functional units that mediate the interactions between the neurons. The most common type of synapse is the *chemical* synapse. In this a presynaptic process liberates a *transmitter* substance that diffuses the synaptic junction between neurons and then acts on a postsynaptic process. Thus a synapse converts a presynaptic electrical signal into a chemical signal and then back into a post synaptic electrical signal (Shepherd and Koch, 1990). In traditional descriptions of neural organization, it is assumed that a synapse is a simple connection that can impose *excitation* or *inhibition*, but not both on the receptive neuron.

A developing neuron is synonymous with a plastic brain: *plasticity* permits the developing nervous system to adapt to its surrounding environment (Churchland and Sejnowski, 1992; Eggermont, 1990). In an adult brain, plasticity may be accounted for by two mechanisms; the creation of new synaptic connections between neurons, and the modification of existing synapses. *Axons*, the transmission lines, and *dendrites*, the receptive zones, constitute the cell filament that may be distinguished on morphological grounds: an axon has a smoother surface, fewer branches, and greater length, whereas dendrites has a irregular surface and more branches (Freeman, 1975). Neurons come in wide variety of shapes and sizes in different parts of the brain. Fig. 4.1 illustrates the shape of a *pyramidal cell* which is one of the most common types of cortical neurons. The input is received through the dendritic spines. The pyramidal cell receives 10,000 or more synaptic contacts and it can project into thousands of target cells.

For neural networks, also, plasticity is essential for its functioning. A *neural network* is a machine that is designed to *model* the way in which the brain performs a particular task or function of interest; the network is usually implemented using electronic components or simulated software on digital computer. This work is confined to neural networks that perform useful computations through the process of learning by employing massive interconnections of simple computing cells called "neurons" or "processing units".

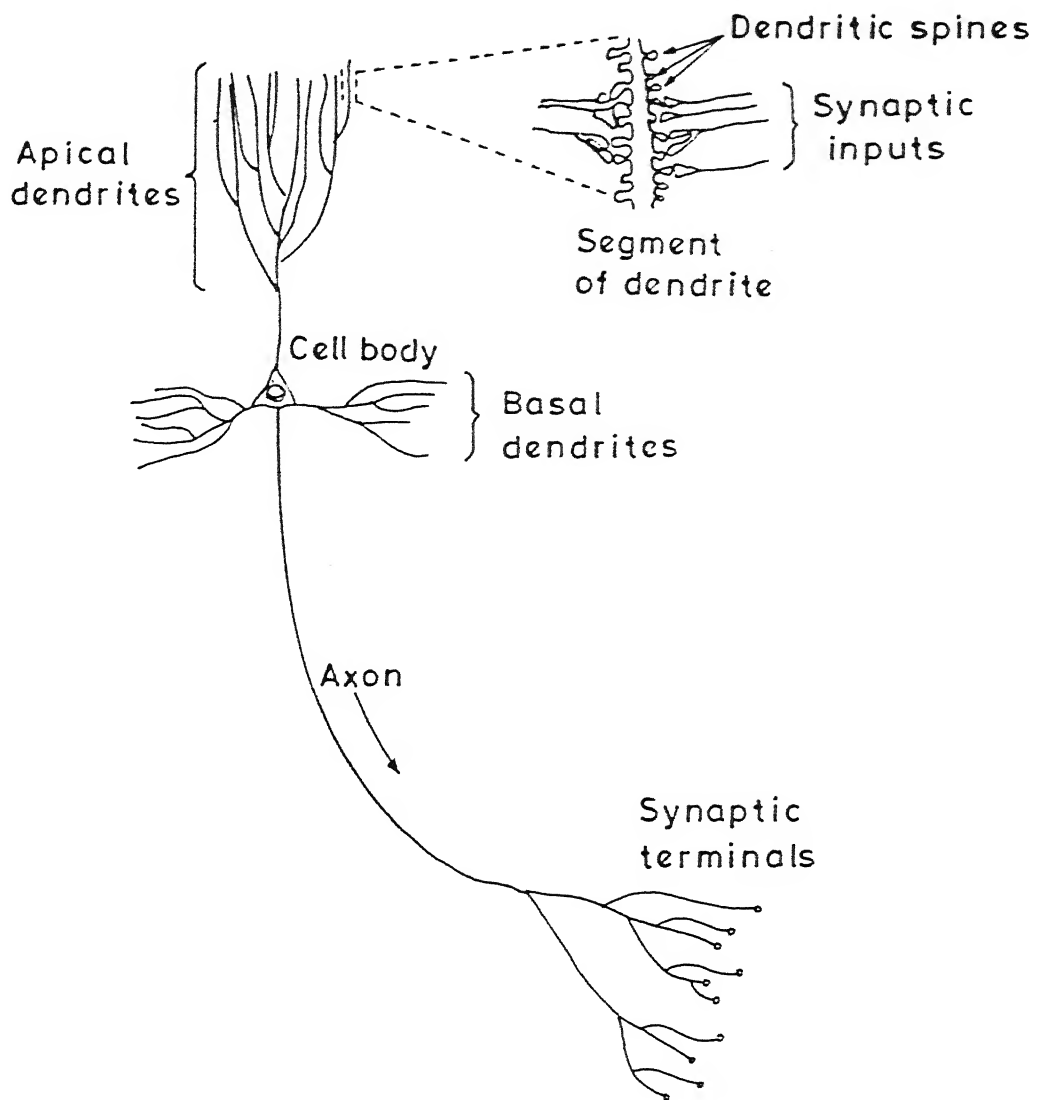


Fig.4.1 The pyramidal cell

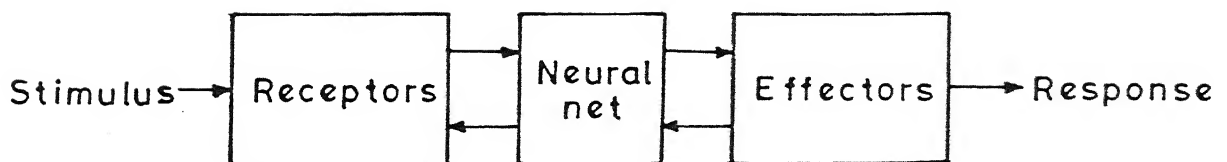


Fig. 4.2 Block diagram representation of nervous system

A definition adapted from Aleksander and Morton (1990) viewing neural network as adaptive machine is as follows:

A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. *Knowledge is acquired by the network through the learning process.*
2. *Interneuron connection strength known as synaptic weights are used to store the knowledge.*

The procedure used to store the knowledge is called a *learning algorithm*, the function of which is to modify the synaptic weights of the network in an orderly fashion so as to attain a desired objective.

Neural Networks are also referred to in the literature as *neurocomputers*, *connectionist networks* and *parallel distributed processors*.

4.1.1 Structural Levels of Organization in the Brain

The human nervous system may be viewed as a three stage system, as depicted in the block diagram shown in Fig. 4.2 (Arbib, 1987). The brain lies in the center of the system which is shown as *neural net* which perpetually receives information, perceives it, and makes appropriate decisions. Two sets of arrows are shown in Fig. 4.2. Those pointing from left to right indicate forward transmission of information-bearing signals through the system. While, the arrows pointing from right to left indicate feedback in the system. The *receptors* in Fig. 4.2 convert the stimuli from the human body or the ambience into electrical impulses that convey information to the neural net (the brain). The *effectors*, on the other hand, convert the electrical impulses generated by the neural net into known responses as system outputs.

An extensive study has been put on the analysis of the local regions of brain (Churchland and Sejnowski, 1992; Shepherd and Koch, 1990). As outcome of this research is the

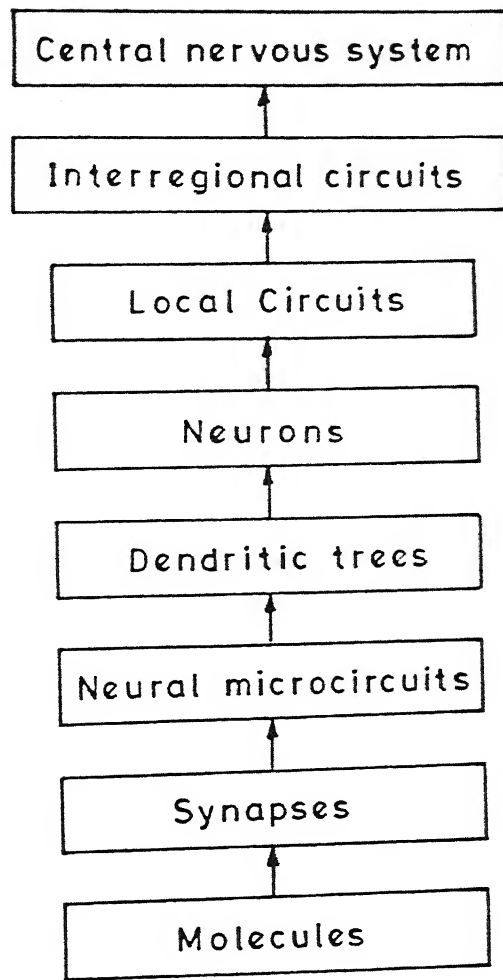


Fig.4.3 Structural organization of levels in the brain

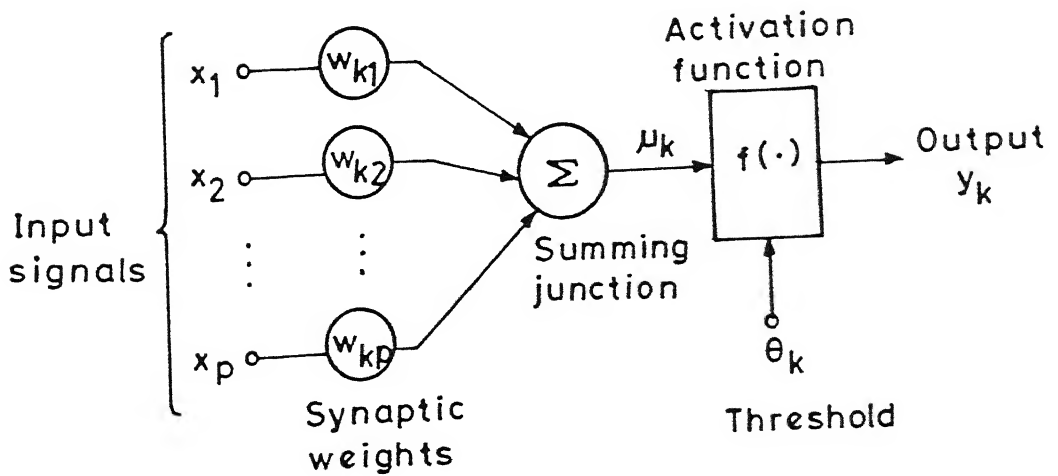


Fig.4.4 Nonlinear model of a neuron

hierarchy of interwoven levels of organization shown in Fig. 4.3. The Fig. shows the basic level as *synapse*, which of course depends on ions and molecules in action. The assembly of these synapses constitutes the *neural microcircuit* organized into patterns of connectivity so as to produce a functional operation of interest. They group to form *dendritic subunits* within the *dendritic trees* of individual neurons. The whole *neuron*, about $100\mu\text{m}$ in size contains several dendritic subunits. At the next level of complexity, we have *local circuits* (about 1mm in size) made up of neurons with similar or different properties; these assemblies perform operations characteristic of a localized region in the brain. This followed by *interregional circuits* made up of pathways, columns, and topographic maps. *Topographic maps* are organized to respond to incoming sensory information. Finally, the topographic maps, and other interregional circuits mediate specific behavior in the *central nervous system*.

4.1.2 Models of a Neuron

A neuron is the information-processing unit that is fundamental to the operation of a neural network. A model of *neuron* is shown in the Fig. 4.4. There may be three basic elements of the neuron:

1. A set of *synapses* or *connecting links*, each of which is characterized by a *weight* or *strength* of its own. Specifically, a signal x_i at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} . The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. The weight is positive if the associated synapse is excitatory; it is negative if the associated synapse is inhibitory.
2. An *adder* for summing the input signals, weighted by the respective synapses of the neuron.
3. An *activation function* for limiting the amplitude of the output of a neuron. The activation function is also referred to in the literature as a *squashing function* in that it limits the permissible amplitude range of the output signal to some finite value.

The model shown in Fig. 4.4 also includes an externally applied *threshold* θ_k that has the effect of lowering the net input of activation function. On the other hand, the net input of

the activation function may be increased by employing the *bias* term rather than a threshold; the bias is the negative of the threshold.

Mathematically the neuron k may be described as follows:

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (4.1)$$

and

$$y_k = f(u_k - \theta_k) \quad (4.2)$$

where x_1, x_2, \dots, x_p are the input signals and $w_{i1}, w_{i2}, \dots, w_{ip}$ are the synaptic weights of the neuron k ; u_i is the *linear combiner output*; θ_k is the threshold; f is the *activation function* and y_k is the output signal of the neuron.

4.1.3 Types of Activation Function

The activation function, denoted by f defines the output of a neuron in terms of the activity level at its input. We may identify three basic types of activation functions:

1. **Threshold Function.** Refer to Fig. 4.5(a) which shows this type of function. It may be expressed as

$$f(v) = 1 \quad \text{if } v \geq 0 \quad (4.3)$$

$$f(v) = 0 \quad \text{if } v < 0$$

Correspondingly, the output of neuron k employing such a threshold function is expressed as

$$y_k = 1 \quad \text{if } v_k \geq 0 \quad (4.4)$$

$$y_k = 0 \quad \text{if } v_k < 0$$

where v_k is the internal activity level of the neuron; that is

$$v_k = \sum_{j=1}^p w_{kj} \theta_k \quad (4.5)$$

Such a neuron is referred in literature as the *McCulloch-Pitts model*, in recognition of the pioneering work done by McCulloch and Pitts (1943). The neuron may take a value of 1 as output if the total internal activity level of that neuron is nonnegative and 0 otherwise.

2. **Piecewise-Linear Function.** for the piecewise-linear function, described in Fig. 4.5(b), we have

$$\begin{aligned} f(v) &= 1 & \text{if } v \geq 0.5 \\ f(v) &= v & \text{if } -0.5 > v > 0.5 \\ f(v) &= 0 & \text{if } v \leq -0.5 \end{aligned} \quad (4.6)$$

where the amplification factor inside the linear region of operation is assumed to be unity. This function may be viewed as an approximation to a nonlinear amplifier.

3. **Sigmoid Function.** The sigmoid function is by far the most common form of activation function used in the construction of artificial neural networks. It is defined as a strictly increasing function that exhibits smoothness and asymptotic properties. An example of the sigmoid is the *logistic function*, defined by

$$f(v) = \frac{1}{1 + \exp(-\alpha v)} \quad (4.7)$$

where α is the *slope parameter* of the sigmoid function (Fig. 4.5(c)).

4.1.4 Learning Process

Among the many interesting properties of a neural network, the property that is of primary significance is the ability of the network to *learn* from its environment, and to improve its performance through learning; the improvement in performance takes place over time in accordance with some prescribed measure. A neural network learns about its environment through an iterative process of adjustments applied to its synaptic weights and thresholds. Ideally, the network becomes more knowledgeable about its environment after each iteration of the learning process. Learning is defined as follows (adapted from Mendel and McClaren (1970)) :

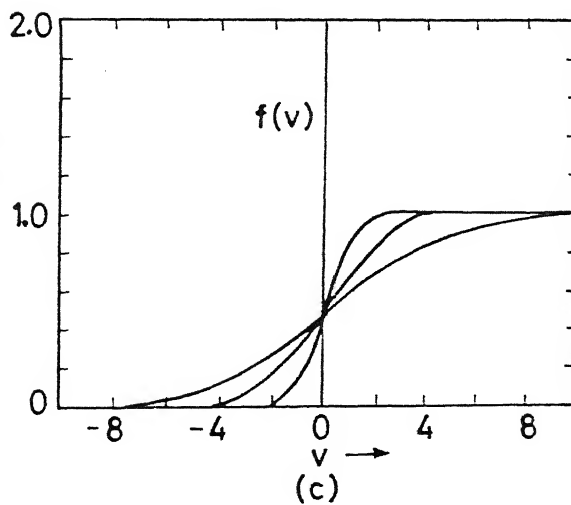
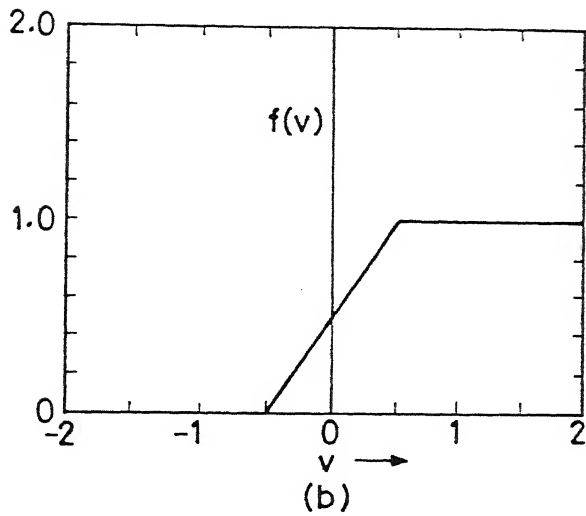
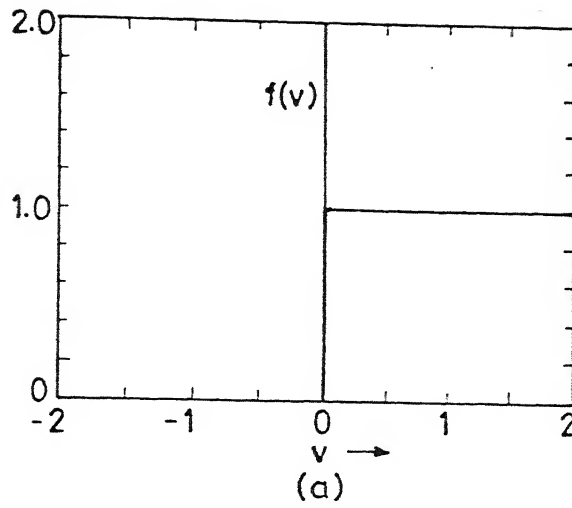


Fig.4.5 (a) Threshold function (b) Piecewise-linear function (c) Sigmoid function

iteration of the learning process. Learning is defined as follows (adapted from Mendel and McClaren (1970)) :

Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. the type of learning is determined by the manner in which the parameter changes take place.

The definition of the learning process implies the following sequence of events:

1. The neural network is *stimulated* by an environment.
2. The neural network *undergoes changes* as a result of this stimulation.
3. The neural network *responds in a new way* to the environment, because of the changes that have occurred in its internal structure.

A prescribed set of well-defined rules for the solution of a learning problem is called a *learning algorithm*. In this work two learning paradigms have been used *supervised* which employs the error-correction learning rule and *unsupervised* using competitive learning rule. These are separately dealt later.

4.1.5 The Perceptron Model

The *perceptron* is the simplest form of a neural network used for the classification of a special type of patterns said to be *linearly separable* (i.e., patterns that lie on a opposite side of a hyperplane). Basically, it consists of a single neuron with adjustable synaptic weights and threshold, as shown in the Fig. 4.6. This single layer neuron is limited for pattern classification with only two classes. The learning procedure for this model was first developed by Rosenblatt (1958, 1962). Basic to the operation of Rosenblatt's perceptron is the *McCulloch-Pitts model* of a neuron. We may recall from earlier discussion that such a neuron model consists of a linear combiner followed by a hard-limiter. Accordingly, the neuron produces an output equal to +1 if the input is positive and zero if it is negative. A single -layer perceptron bears a close relationship with the *Gaussian Maximum Likelihood Classifier*, in that they are both are examples of linear classifiers (Duda and Hart, 1973). As discussed earlier in Chapter 3 this particular

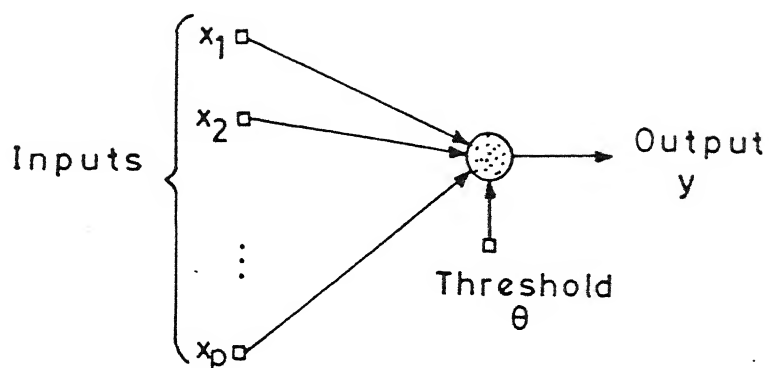


Fig.4.6 Single layer perceptron

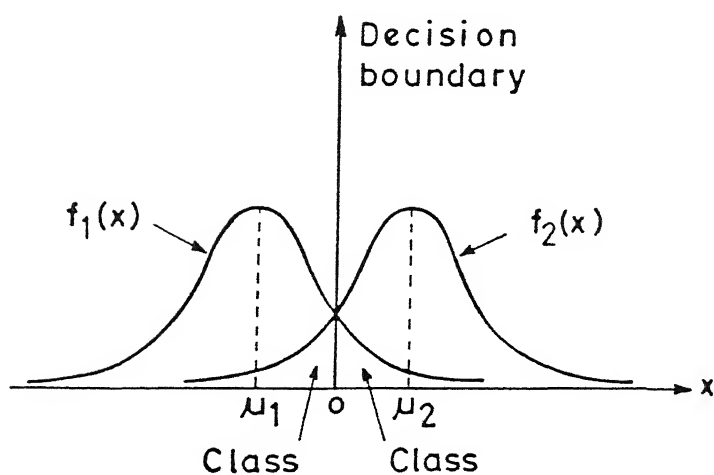


Fig.4.7 Two overlapping, one-dimensional Gaussian distributions

classifier is a parameter-estimation method which assumes the parameters as fixed and unknown values. The best estimate is defined to be the one that maximizes the probability of obtaining samples actually observed. The joint probability distribution earlier obtained is given by

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} (\det C)^{1/2}} \exp[-1/2(\mathbf{x} - \boldsymbol{\mu})^T C^{-1}(\mathbf{x} - \boldsymbol{\mu})] \quad (4.8)$$

where C is the determinant of the covariance matrix C , \mathbf{x} is a Gaussian-distributed vector and p is the dimension of vector \mathbf{x} .

For our case of comparison let's consider a two class problem having a vector \mathbf{x} given as follows, the two classes being C_1 and C_2 :

- $\mathbf{x} \in X_1$: mean vector = μ_1
 covariance matrix = C
- $\mathbf{x} \in X_2$: mean vector = μ_2
 covariance matrix = C

The vector \mathbf{x} has a mean vector the value of which depends on its class membership, and the covariance matrix is same for both. We may further assume that:

- Both the classes are equiprobable
- The samples of both the classes are correlated so that the covariance matrix is nondiagonal
- The covariance matrix is nonsingular such that the inverse of the matrix exists.

Now, for the purpose of pattern classification considered here, we may define a *log likelihood* as follows:

$$l_i(\mathbf{x}) = \boldsymbol{\mu}_i^T C^{-1} \mathbf{x} - 0.5 \boldsymbol{\mu}_i^T C^{-1} \boldsymbol{\mu}_i \quad \text{where } i = 1, 2.$$

Thus, on performing $l = l_1(\mathbf{x}) - l_2(\mathbf{x})$ we get

$$l = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T C^{-1} \mathbf{x} - 0.5(\boldsymbol{\mu}_1^T C^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T C^{-1} \boldsymbol{\mu}_2)$$

which is *linearly* related to vector \mathbf{x} .

The pattern-classification problem, now, can be solved using the following rule:

if $l \geq 0$, then $l_1 \geq l_2$ and therefore assign \mathbf{x} to class C_1

if $l < 0$, then $l_2 > l_1$ and therefore assign \mathbf{x} to class C_2

Thus, the operation of the Gaussian classifier is analogous to that of the single-layer perceptron in that they are both classifiers. However, there are certain differences between the two, that are stated below:

- The general principle on which the single layer perceptron acts is that the patterns to be classified are *linearly separable*. The Gaussian distribution of the two patterns assumed in the derivation of the maximum-likelihood Gaussian classifier do certainly overlap and therefore are not exactly separable; the extent of overlap is determined by the mean vectors μ_1 and μ_2 , and the covariance matrices C_1 and C_2 . The overlapping for a 1-D Gaussian case is illustrated in Fig. 4.7. When the inputs are not separable and their distributions overlap, the perceptron would develop a problem, in that the decision boundaries may oscillate continuously (Lippman, 1987).
- The maximum likelihood classifier minimizes the average probability of classification error. This minimization is independent of the overlap between the underlying gaussian distributions of the two classes.
- The perceptron convergence algorithm is *nonparametric* in the sense that it makes no assumption concerning the form of the underlying distributions; it operates by concentrating on the errors that occur where the distributions overlap. It may thus be more robust than classical techniques, and work well when the inputs are generated by nonlinear physical mechanisms, and whose distributions are heavily skewed and non-

Gaussian (Lippman, 1987). In contrast, the maximum likelihood Gaussian classifier is *parametric* ; its derivation assumes underlying distribution as Gaussian, which may limit its application.

- The perceptron convergence algorithm is both adaptive and simple to implement; its storage requirement is confined to the set of synaptic weights and threshold. The design of MLE is fixed and can be made adaptive only at the cost of increasing the storage requirement and more complex computations(Lippman, 1987).

Till now a single neuron perceptron model has been discussed. Now we study an important class of neural networks, namely, multilayered feedforward networks is investigated. These networks consist of *input layer*, one or more *hidden layers* of computation node, and an *output layer* of computation nodes. These are called multilayer perceptrons. Multilayer perceptrons have been successfully used in past to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the *error back propagation algorithm* which uses the *error correction learning rule*. The learning process performed with this algorithm is called the *back propagation learning* (Schalkoff, 1992).

Multilayer perceptrons have three distinctive characteristics:

1. The model of each neuron in the network includes *nonlinearity* at the output end. This non linearity should be differentiable everywhere, As opposed to the hard limiting used in Rosenblatt's perceptrons. A commonly used form of non linearity that satisfies this requirement is the *sigmoidal non linearity* (discussed later).

2. The network contains one or more layers of *hidden neurons* that are not part of input output part of network.
3. The network exhibits high degree of *connectivity*, determined by the synapses of the network.

Fig. 4.8 shows the architectural graph of a multilayer perceptron with two hidden layers. The network shown here is sufficiently general in nature, each neuron in any layer is connected to all the nodes/neurons in the previous layer. Each hidden or output neuron of a multilayer perceptron is designed to perform two computations:

1. The computation of the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the input signals and synaptic weights associated with that neuron.
2. The computation of an instantaneous estimate of the gradient vector (i.e., the gradients of the surface with respect to the weights connected to the inputs of a neuron), which is needed for the backward pass (discussed later) through the network.

4.1.6. Backpropagation Algorithm

4.1.6.1 General

An artificial neural network is constructed from a set of processing units interconnected by weighted channels according to some architecture. Each unit consists of a number of input channels, an activation function, and an output channel. Signals impinging on inputs of an unit are multiplied by the channels weight and are summed to derive the net input to that unit. The net unit is then transformed by the activation function f to produce an output for the unit (Fig. 4.9). This may be expressed as

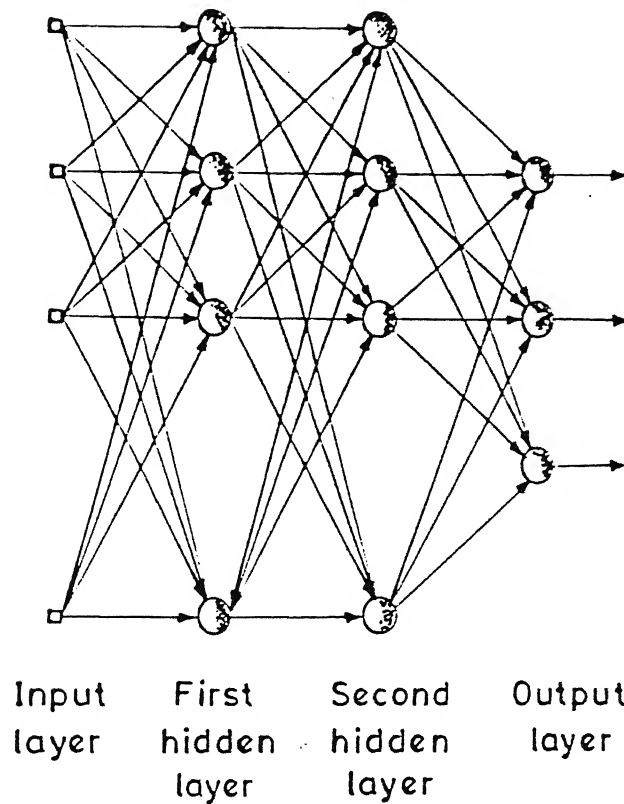


Fig.4.8 Architectural graph of a multilayer perceptron with two hidden layers

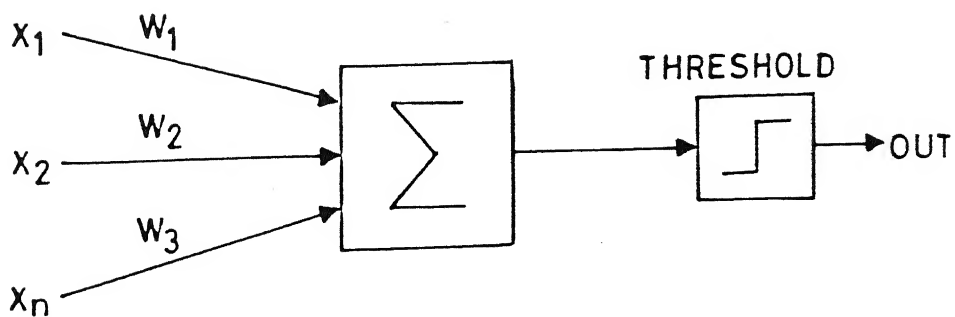


Fig.4.9 Perceptron Neuron

$$NET = \sum_{i=1}^n O_i W_i \quad (4.9)$$

$$OUT = f(NET) \quad (4.10)$$

where O_i is the magnitude of the i input and W is the weight of the interconnection channel. Backpropagation is a systematic method for training multilayer artificial neural networks. It has a mathematical foundation that is strong if not highly practical. Fig. 4.9 shows the neuron used as the fundamental building block for backpropagation networks. The activation function f used in this work is the sigmoidal function, given earlier by Equation 4.7.

This is the sigmoidal function (fig. 4.5(c)). It is desirable in that it has a simple derivative, a fact we use in implementing the backpropagation algorithm.

$$\frac{\partial OUT}{\partial NET} = OUT(1 - OUT) \quad (4.11)$$

Sometimes called a logistic, or simply a squashing function, the sigmoid compresses the range of NET so that OUT lies between 0 and 1. This squashing function also introduces the non-linearity required for greater representational power. This function is also differentiable everywhere which is the prime requirement of this algorithm. It has an advantage to provide an automatic gain control. For small signals (NET near zero) the slope of the input/output curve is steep, producing high gain. As the magnitude of the signal becomes greater the gain decreases. Thus, large signals can be accommodated by the network without getting saturated and small signals pass without attenuation. In this work a multilayered network having one hidden layer with three neurons in each layer, has been used. In the present work the convention of defining the hidden layers as the layers of neurons without input layer and the output layer has been used.

4.1.6.2 Training

The objective of training the network is to adjust the weights so that the application of a set of inputs produces the desired set of outputs. These sets of inputs and outputs are called vectors. In this study the input is a vector having four elements, that is the pixel values of the image in four bands. There is one output node. Gray level values are scaled and so is the output node. At the output node the target value assigned and expected is the class number. Before the training process is started all the values are initialized to small random numbers. This would avoid the network to get saturated with large weight values. Training of the backpropagation requires the following steps:

- Select the next training pair and apply the input vector to the network
- Calculate the output of the network
- Calculate the error between the expected output and the obtained output.
- Adjust the weights of the network in a way such that the error is minimized.
- Repeat the above steps till the error is acceptably low.

The operations required in steps 1 and 2 above are similar to the way in which the trained network will ultimately be used; that is, an input vector is applied and the resulting output is calculated. Calculations are performed on a layer-by-layer basis. In step 3, each of the network outputs is subtracted from its corresponding component of the target vector to produce error. This error is used in step 4 to adjust the weights of the network, where the polarity and magnitude of the weight changes are determined by the training algorithm. After enough repetitions of these four steps, the error between actual outputs and target outputs should be reduced to an acceptable value, and the network is said to be trained. At this point, the network is used for recognition and weights are not changed.

It may be seen that steps 1 and 2 constitute a "forward pass" in that the signal propagates from the network input to its output. Steps 3 and 4 are a "reverse pass", here, the calculated error signal propagates backward through the network where it is used to adjust weights.

4.1.6.3 Forward Pass

In the steps 1 and 2 the input vector pair X and T comes from the training set and calculation is performed on X to produce the output Y . The calculation in the multilayer network is done layer by layer, starting at the layer nearest to the inputs. So if the weight vector of the neurons is considered as W then the NET vector can be written as $N=XW$. On applying the activation function f the output at each neuron can be written as $O=f(N)$. The output of one layer is input to another.

4.1.6.4 Reverse Pass

A little modification of delta learning rule accomplishes this task. The error is multiplied by the derivative of the squashing function $[OUT(1 - OUT)]$.

The following equation illustrate the calculation.:

$$\delta = OUT (1 - OUT) (TARGET - OUT) \quad (4.12)$$

$$\Delta w_{pq,k} = \eta \delta_{q,k} OUT_{pj} \quad (4.13)$$

$$w_{pq,k} (n+1) = w_{pq,k}(n) + \Delta w_{pq,k} \quad (4.14)$$

where

$w_{pq,k}(n)$ is the value of a weight from a neuron p in the hidden layer to a neuron q in the output layer at step n (before adjustment); note that the subscript k indicates that the weight is associated with its destination layer.

$w_{pq,k} (n+1)$ is the value of the weight at step $n+1$ (after adjustment)

$\delta_{q,k}$ is the value of δ for neuron q in the output layer k and

OUT_{pj} is the value of OUT for neuron p in the hidden layer j .

4.1.7 Kohonen's Self Organizing Feature Maps

One important organizing principle of sensory pathways in the brain is that the placement of neurons is orderly and often reflects some physical characteristic of the external stimulus being sensed (Kandel and Swartz, 1985). Although much of the low level

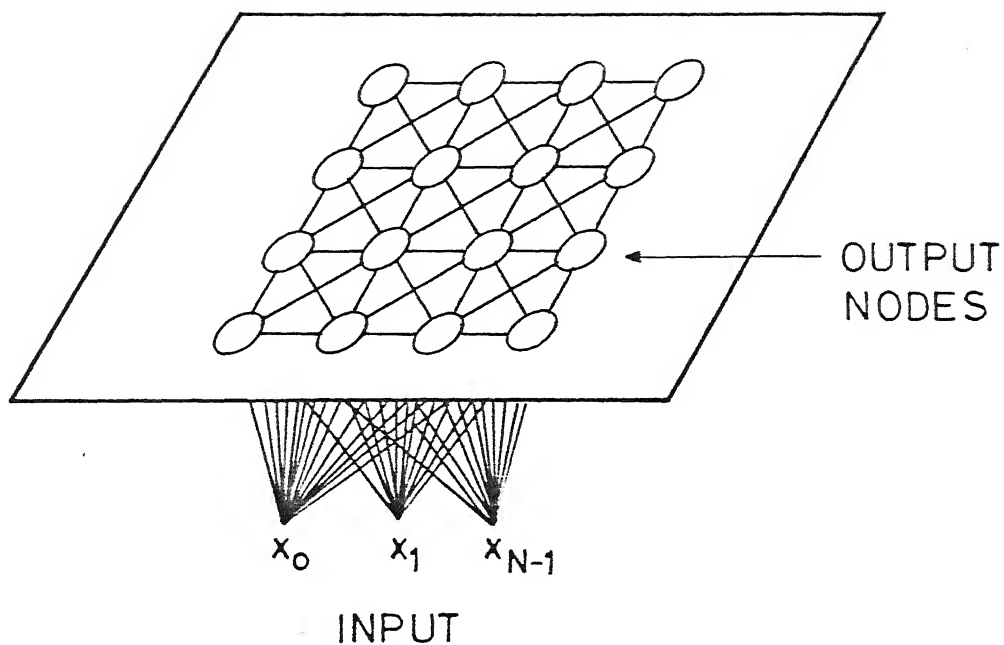


Fig. 4.10 Two-dimensional array of output nodes used to form feature maps. Every input is connected to every output node via variable connection weight.

organization is generally pre-determined, it is likely that some of the organization at higher levels is created during learning by algorithms which promote self-organization. Kohonen presents one such algorithm which produces what he calls self-organizing feature maps similar to those that occur in brain

Kohonen's algorithm creates a vector quantizer by adjusting weights from common input nodes to M output nodes arranged in a 2-D grid as shown in the Fig. 4.10. Output nodes are extensively connected with many local interconnections. Continuous-valued input vectors are presented sequentially in time without specifying the desired output. After, enough vectors have been presented, weights will specify cluster or vector clusters that sample the input space such that the point density function of the vector clusters tend to approximate the probability density function of the input vectors (Kohonen, 1984). In addition, the weights will be organized such that topologically close nodes are sensitive to inputs that are physically similar. Output nodes will thus be ordered in a natural manner. In the algorithm given on the next page, it is clear that this algorithm requires choosing of several parameters such as learning rate and size of an update neighborhood. The essential ingredients of this algorithm can be summarized as follows:

- A one or two-dimensional lattice of neurons that computes simple discriminant functions of inputs received from an input of arbitrary dimension.
- A mechanism that compares these discriminant functions and selects the neuron with the largest discriminant function value.
- An interactive network that activates the selected neuron and its neighborhood simultaneously.
- An adaptive process that enables the activated neurons to increase their discriminant function values in relation to the input signals.

To proceed with the development of the algorithm, consider Fig. 4.9, which depicts a two dimensional lattice of neurons. The input vector, representing the set of input signals, is denoted by

$$\mathbf{x} = [x_1, x_2, \dots, x_p]^T$$

The synaptic weight vector of neuron j is denoted by

$$\mathbf{w}_j = [w_{j1}, w_{j2}, \dots, w_{jp}]^T, j = 1, 2, \dots, N$$

The best matching criterion is the selection of minimum Euclidean distance between vectors. Specifically, if the index $i(\mathbf{x})$ is used to identify the neuron that best matches the input vector \mathbf{x} , $i(\mathbf{x})$ is determined by applying condition

$$i(\mathbf{x}) = \arg \min ||\mathbf{x} - \mathbf{w}_j||, \quad j = 1, 2, 3, \dots, N$$

where $||\cdot||$ denotes the Euclidean norm of the argument vector.

Algorithm for Self-Organizing Feature Maps

Step 1. Initialize weights

Initialize weights from N inputs to the M output nodes (as shown in Fig. 4.10) to small random values. Set the initial radius of the neighborhood as shown in Fig. 4.11

Step 2. Present new input

Step 3. Compute distance to all nodes

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2$$

where $x_i(t)$ is the input to node i at time t , w_{ij} is the weight from input node i to output node time t .

Step 4. Select output node with minimum distance

Select node j^* as that output node with minimum distance d_j as shown in Fig. 4.11.

Step 5. Update weights to node j^* and neighbors

Weights are updated for node j^* and all nodes in the neighborhood defined by Ne_{j^*} as shown in Fig. 4.11. New weights are

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) (x_i(t) - w_{ij}(t))$$

4.2 Methodology for Classification

4.2.1 General

We have discussed the theoretical aspects of Neural Networks. We now focus on the practical implementation of our objectives in this work. The study involved, firstly, selection of training area and using it to train the various classifiers. A test area was also selected which was used to test the accuracy of prediction of the trained classifiers. Finally, the entire image was classified using trained network. The training area used to train the network is the same as the one used for the statistical pattern classifier in section 3.3.1. The selection of training area has also been discussed in detail in section 3.3.1.

4.2.2 Classification using Backpropagation learning

The Backpropagation learning algorithm involves consideration of various parameters to get successfully trained and then generalize the results. The Backpropagation learning is shown in Fig. 4.12. Various parameters used are:

1. Input pattern
2. Type of Non-Linearity
3. Desired Output pattern
4. Number of Hidden layers
5. Stopping criteria
6. Maximum desired error
7. Bias
8. Learning rate parameter
9. Momentum factor
10. Logistic gain

1. Input Pattern

The input pattern consisted of the pixel values in the four bands of the IRS-1B data. Thus, four input neurons selected. The inputs consisted of Pixel values in the four bands. The network scaled the values of these inputs between 0.1-0.9 (to avoid saturation). The network receives these values pixel by pixel, to perform further computations. The role of input layer is somewhat fictitious, in that input layer 'holds' these values and distribute them to all units in the next layer. Thus, the input layer units do not implement a separate mapping or conversion of the input data, and their weights are insignificant.

2. Desired Output Patterns

As discussed earlier, the feedforward networks should have the ability to learn pattern mappings. Once the patterns to be classified are presented to the network, it performs computations and gives an output. The output received in this study are labels of various known classes. They were normalized between 0.1-0.9. For level -I classification six output neurons and thus six different output patterns each denoting a class, as shown on the next page:

| Output Pattern | Class |
|----------------|-------|
| 1 0 0 0 0 0 | 1 |
| 0 1 0 0 0 0 | 2 |
| 0 0 1 0 0 0 | 3 |
| 0 0 0 1 0 0 | 4 |
| 0 0 0 0 1 0 | 5 |
| 0 0 0 0 0 1 | 6 |

For level - II classification eight output neurons were chosen resulting in eight output patterns, each pattern representing a class as shown below:

| Output Pattern | Class |
|-----------------------|--------------|
| 1 0 0 0 0 0 0 0 | 1 |
| 0 1 0 0 0 0 0 0 | 2 |
| 0 0 1 0 0 0 0 0 | 3 |
| 0 0 0 1 0 0 0 0 | 4 |
| 0 0 0 0 1 0 0 0 | 5 |
| 0 0 0 0 0 1 0 0 | 6 |
| 0 0 0 0 0 0 1 0 | 7 |
| 0 0 0 0 0 0 0 1 | 8 |

3. Number of Hidden Layers

There are several interesting and related interpretations that may be attached to the units in the internal layers (the internal units). The internal layers remap the inputs and the results of previous internal layers to achieve a pre classifiable representation of the data. As the number of units to be taken are concerned, it is in fact a very difficult question to answer. We have selected a single hidden layer network, a two layer network and a three layer network for both the classifications separately. This is done for the sake of comparison and to arrive at an optimal at an optimal topology of the network. Obviously, the computation time increases when the number of layers are increased.

4. Type of Non Linearity

The sigmoidal non linearity is selected as the activation functions. This is most commonly used function due to its differentiable (hence continuous) nature. This is the only criteria for the selection of the non-linearity. The logistic function used has been described in chapter four.

5. Stopping Criteria

Computation has been performed for various iterations, 500, 1000, 5000, 10000, 15000 and 20000 for level - I classification. For level -II classification, computation was performed for 500,1000,5000,10,000,20,000,30,000 and 50,000 iterations. For Remote Sensing purposes, the final error in the value of the labels is not taken as the judging criteria. Instead, the percentage accuracy in the final classification accuracy is the stopping criteria. Various iterations are tried for the sake of comparison and deriving at the optimum number of iterations having the highest accuracy. The increasing number of iterations results in an increase in the computation time of the network.

6. Rate of Learning

The Backpropagation provides an approximation to the trajectory in the weight space computed by the method of steepest descent. The smaller is the learning parameter, the smaller will be the changes to the synaptic weights in the network from the one iteration to the next and smaller will be the trajectory in the weight space. This results in a very slow rate of learning. On increasing the value, speed of computation increases, but the synaptic weights may assume oscillatory nature. Hence, after trying various values 0.8 was taken as the optimum value in level - I classification. However, in level - II classification the value for the learning rate parameter were chosen as 0.0001.

7. Logistic Gain:

This determines the slope of the sigmoidal non linearity. A value of 0.75 was selected. .

8. Momentum Factor

The momentum term is added as it may prevent oscillations and may help the system to escape local minima of the error function in the training process. A value of 0.8 for the momentum factor has been selected for level - I classification and 0.005 for level - II classification.

9. Neuron Bias

A bias neuron value of 1 is added to accelerate the convergence process. It actually offsets the origin of the logistic function, producing an effect that is similar to adjusting the threshold of the perceptron neuron.

10. Training and Testing

The network was trained for about 120 samples and tested for 300 samples, for level -I classification. For level-II classification the network was trained for 160 samples and tested for 400 samples. Finally, the entire image was classified with the trained network.

4.2.3 Classification Using Kohonen's Self Organizing Maps

This is an unsupervised neural network approach to classification and results in formation of clusters. Thus, the training file includes the data of four bands. Pixel by pixel, input is presented to the network, sequentially with time, without specifying the desired output. We randomize the weights using random number generator in C programming language. As required by the algorithm, this value is extremely small. Hence, we may make them comparable to our input data either by normalizing the input or by multiplying the weights by some finite value. We have used the second approach, and have named the multiplication factor as the *orientation factor*. This has been selected as 38 after several trials. Further, we have worked on the rectangular topology only for the output neurons. We have selected the initial radius as the

$[n]$ ($[]$ denotes the *greatest integral function*, n is the number of output nodes). The number of output nodes, n , is taken as six. The gain term is selected initially as 0.1 and it decreases with time. The training file consisted of about 120 pixel values and the test file of 300 values. The trained network is then applied to the image to generate a classified output.

4.3 Methodology for river extraction

For river extraction, the ANN using Backpropagation learning was trained for almost same network parameters as that of level - I classification except that the output pattern was now taken as 0 or 1. A training data set comprising of 100 different pixel values belonging to the river were taken. The fourth band was selected for this and the desired value assigned in the training set was 1. Finally, when the network was trained, the image in fourth band was presented to the network. Those outputs obtained between 0.0 and 0.5 were taken as 0 while the outputs with values between 0.5 and 1.0 were taken as 1. This has been done to avoid features other features than river from the image.

4.4 Results

4.4.1 Back Propagation learning

4.4.1.1 Level - I classification

The Back propagation network was trained after various trials on the values of various parameters. Representation of the output was the most important decision to be taken in the process. Initially, only one output neuron was used. The desired output value in this case was the value of the class label itself (scaled between 0.1-0.9). However, the network faced problems in converging for classes 2 and 4 or 3 and 5. Hence, another representation was tried and was adapted as it was found to be appropriate. In this representation, as discussed earlier, the number of output neurons is selected to equal the number of classes, and only one output neuron is active, (i.e., has the value 1). This particular representation has the advantage that only one neuron should be active and all of the others should be inactive. Therefore, the 'winner take all' principle can be used. Thus, during testing an input sample can be classified to the class which has the largest output response (of course within the desired error range). This increases the dimensionality of the network slightly, however, since the desired error range is as high as

50 percent hence the network is able to converge. The number of hidden layers were decided after testing the network on three different sets of hidden layers. It was found

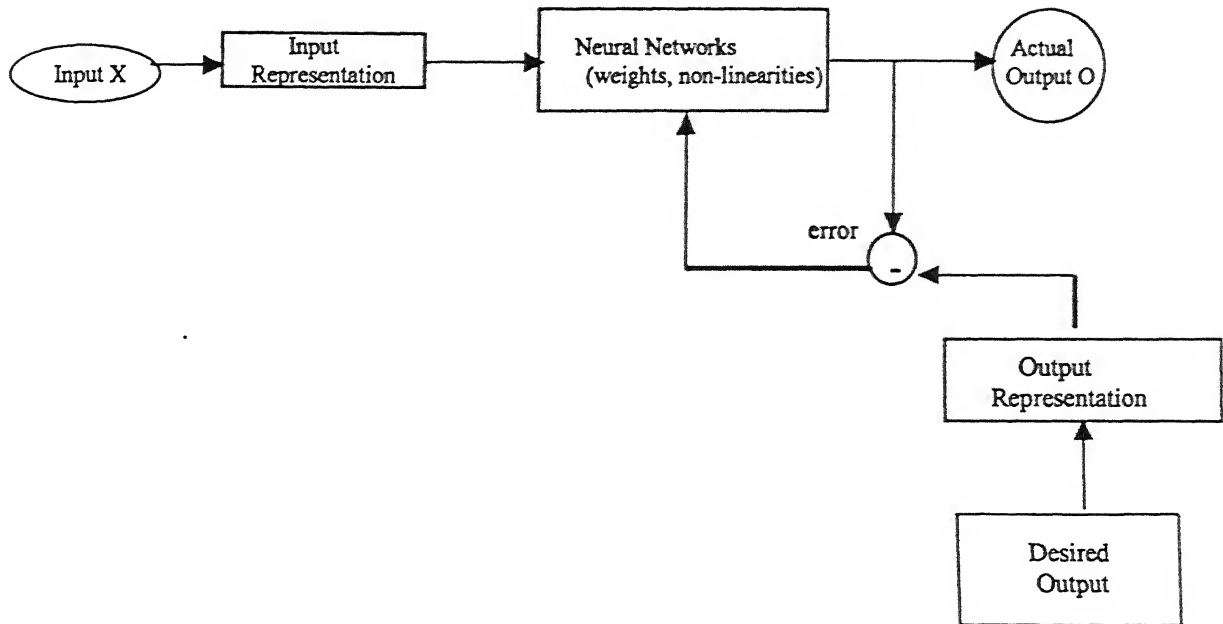


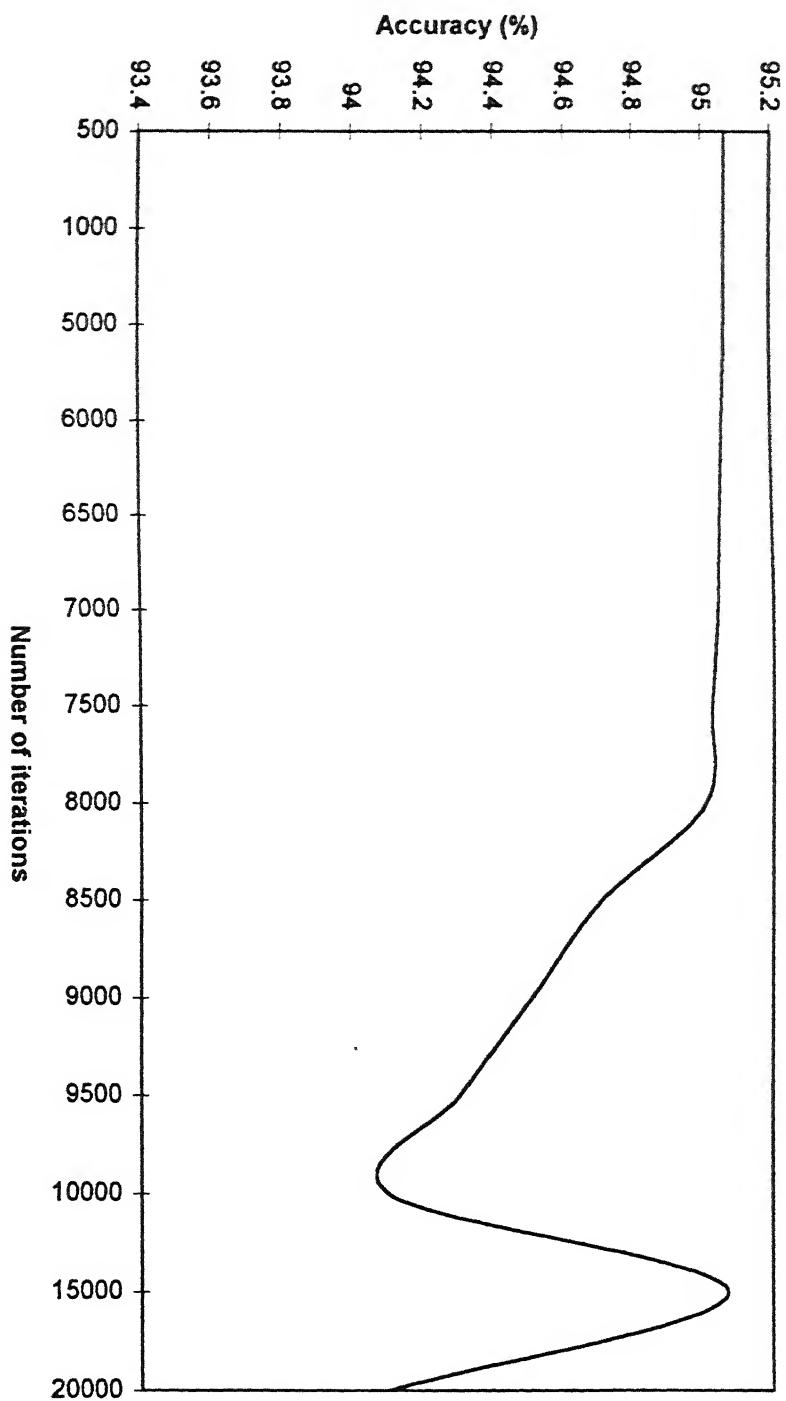
Fig. 4.12 Schematic diagram of Backpropagation Training procedure

that when hidden layers were increased to two and three the network performance deteriorated considerably and to the point of non convergence. However, when the network had one hidden layer, it gave satisfactory results. The results so obtained are summarized in the Tables 4.1-4.6. It will be observed that on using this particular topology of the network, the maximum testing accuracy is reached at 95.07 percent. On increasing the number of iterations, it is observed that the accuracy does not change upto 5000 iterations and starts decreasing on further increasing the iterations (Fig. 4.13). Thus, network could give a fairly high accuracy at 500 iterations. The reason for such low number of iterations is that a high value of learning rate parameter is used which actually accelerates convergence. Normally, a high learning rate parameter is accompanied by oscillations in the network and the network is expected to be caught in a local minima. However, due to selection of high momentum rate parameter this problem does not occur.

Further, as earlier stated, we had not gone for absolute error as the stopping criterion, thus reducing the number of cycles required for training. The misclassifications occur in the case of vegetation and coal dumps. The vegetation (class 3) is classified to an accuracy of 74 percent while coal dumps (class 6) to an accuracy of 98 percent. The rest of the classes, namely, clear water (class 1), turbid water(class 2) and coal mines (class 5) are classified to an accuracy of 100 percent. Fig. 4.13 shows the accuracy versus number of iteration relationship. One can safely conclude that on increasing the number of iterations beyond a certain limit would decrease the efficiency of the ANN classifier. In this case 5000 is that optimum limit. It is anticipated that on changing the parameters of the network one may obtain still better accuracy.

The 512×512 image of the Singrauli area was also tested on this network. The classified image is shown in Fig. 4.14. The image indicates clear water body in H-7 and turbid water in between H-7 and G-7. The yellowish patches in the image are rocky regions while the greenish ones indicate vegetation. The dark red patches like the one in G-5 are the coal dumps. The lighter red patches are of coal mines. The region between A-7 and C-7 is full with green patches and yellow ones. Essentially, this is a vegetated region and has been misclassified as rocks (yellow patches) by the network. Another interesting feature of the image is the region between A-5 and B-1. This region is full of black and blue patches which indicates presence of water bodies. However, this entire region is rocks. There is a possibility of moist zone due to Bijul river or ponds. The misclassification in the image is probably due to the fact that, while the test data was presented to the network sequentially and not randomly that is to say that the test pixels of each class were fed to the network one after the other, in the actual image this was not the case as the pixels came randomly and hence affected the result. Yet another reason could be lack of proper representation of these areas in the training set. This calls for increasing the number of training samples and also careful picking of pixels such that each class is adequately represented in the training data set.

Fig. 4.13 Accuracy Vs Number of iterations (%)



4.4.1.2 Level -II classification

The classes identified in level - I classification were further subdivided into subclasses resulting in eight classes in all. These were clear water, turbid water, sparse vegetation, dense vegetation, quartzite rocks, sandstone, coal dumps and coal mines. The network topology and various parameters were kept the same as that of level-I classification, initially. However, the network did not converge. Thus, a new set of parameters were selected after several trials. The network had shown quite a bit of oscillation with the old set of parameters indicating that it was trapped in a local minima. Thus, the learning rate parameter was drastically reduced to 0.0001. This naturally resulted in increased number of iterations. The network finally converged at 50,000 iterations. The results so obtained are summarized in form of confusion matrix in Table 4.7. The overall accuracy of the classifier comes out to be 95.5 percent. Clear water and turbid water are classified to an accuracy of 100 percent. The sparse vegetation is classified to an accuracy of 72 percent having misclassifications in dense vegetation (class 4), coal mines (class 5) and sandstone (class 7). The dense vegetation (class 4), coal dumps (class 5) and coal mines (class 6) have been classified to 100 percent accuracy. The sandstone region attains an accuracy of 92 percent with misclassifications in dense and sparse vegetation. Finally, the quartzite has misclassification in sparse vegetation and coal dumps having an overall accuracy of 94 percent. The classified image of level - II classification is shown in Fig. 4.15. In the image, reddish patches indicate dense vegetation while the yellow patches are of coal mines. The coal dumps are shown in green. the purple patch in G-7 is of turbid water while the clear water is shown in black in H-7. The dark blue shade scattered in the image indicates sandstone. One may observe these patches in high concentration in H-4. The light blue color present in the image is for quartzite rock. Finally, the white patches indicate the sparse vegetation. One may observe the presence of sparse vegetation in C-4 which is actually quartzite zone, which is actually misclassification as earlier noticed in the test area. Similarly, dark blue patches in G-7 indicating presence of sandstone is also misclassification as this particular region is sparsely vegetated.

Table 4.1 Confusion Matrix for 500 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 37 | 13 | 0 | 0 | 74 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy=95.07 %

Table 4.2 Confusion Matrix for 1000 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 37 | 13 | 0 | 0 | 74 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy=95.07 %

Table 4.3 Confusion Matrix for 5000 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 37 | 13 | 0 | 0 | 74 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy=95.07 %

Table 4.4 Confusion Matrix for 10000 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy (%) |
|---|----|----|----|----|----|----|---------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 36 | 14 | 0 | 0 | 72 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 1 | 0 | 50 | 1 | 96 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy=94.10 %

Table 4.5 Confusion Matrix for 15000 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 37 | 13 | 0 | 0 | 74 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 1 | 0 | 49 | 98 |

Overall Accuracy=95.07 %

Table 4.6 Confusion Matrix for 20,000 iterations in one hidden layer

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy (%) |
|---|----|----|----|----|----|----|---------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 34 | 16 | 0 | 0 | 68 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 100 |
| 6 | 0 | 0 | 0 | 0 | 1 | 49 | 98 |

Overall Accuracy=94.10 %

Table 4.7 Confusion matrix for level -II classification by Backpropagation algorithm

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Accuracy (%) |
|---|----|----|----|----|----|----|----|----|---------------|
| 1 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| 2 | 0 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| 3 | 0 | 0 | 36 | 2 | 4 | 0 | 8 | 0 | 72 |
| 4 | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 0 | 100 |
| 5 | 0 | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 100 |
| 6 | 0 | 0 | 0 | 0 | 0 | 50 | 0 | 0 | 100 |
| 7 | 0 | 0 | 2 | 2 | 0 | 0 | 46 | 0 | 92 |
| 8 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 47 | 94 |

Overall Accuracy = 95.5 percent

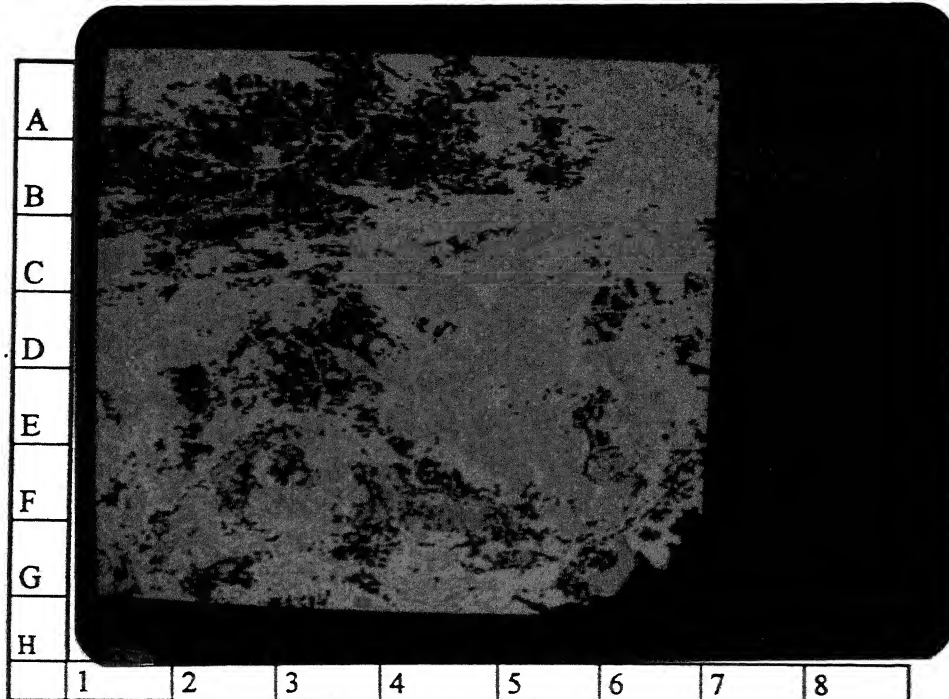


Fig 4.14 Level - I classified image by Backpropagation algorithm

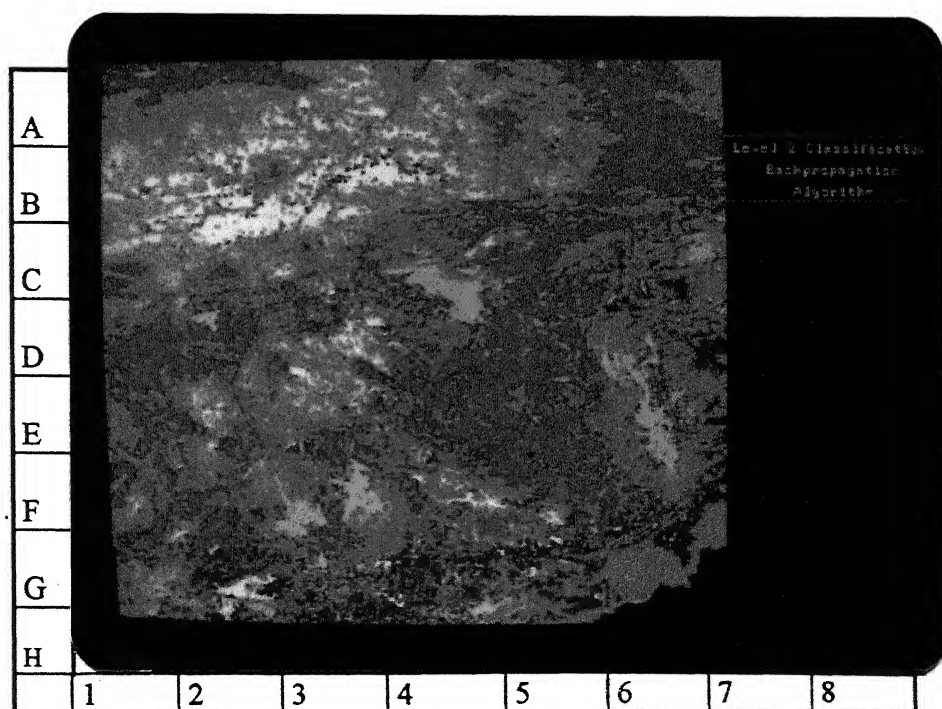


Fig 4.15 Level - II classified image by Backpropagation algorithm

4.2 Kohonen's algorithm

This is an unsupervised classification procedure. The algorithm results in clustering of the test area and the image. It may be first trained using the training set and then used to predict classes for the test area and the image. The results obtained from its clustering are shown in Table 4.8. The overall accuracy obtained is 90.75 percent. The clear water is classified upto an accuracy of 96 percent since some pixels get classified as turbid water. The turbid water is classified with an accuracy of 98 percent as some of its pixels get classified as coal dumps. Vegetation is perfectly well classified having an accuracy of 100 percent. The poorest classification is shown by the rocks having an accuracy of 60 percent. Its pixels are incorrectly classified as coal mines, turbid water and clear water. The coal mines are classified with an accuracy of 76 percent with misclassifications as vegetation, turbid water and coal dumps. The coal dumps are classified with an accuracy of 96 percent with misclassifications as clear water. The classified image of Kohonen's algorithm is shown in Fig. 4.16. The effect of misclassification, noticed in the test area, is evident from the image. A portion of clear water in H-6 and H-7 is classified as turbid water. Similarly, the turbid water (white, G-7) has a blue patch indicating the presence of coal dumps which are shown by blue patches in the image. These coal dump patches like in G-5 are surrounded by some white patches (turbid water), indicating misclassification. The green color is of vegetation. Coal mines are shown in black. However, in certain regions such as the region in F-7 and E-7 characterised by blue and white is a total misclassification of coal mine. These coal mines are misclassified as coal dumps. Another such coal mine lying in D-5 is shown completely green hence being misclassified as vegetation. The rocks are shown in dark purple color and have been misclassified massively. For instance the region between A-1 and B-5 is mostly classified as water and coal dumps while it is actually rock. Thus, the accuracy of the classifier seems to have dropped when the entire image was tested.

Table 4.8 Confusion Matrix of Kohonen's Output

| | 1 | 2 | 3 | 4 | 5 | 6 | Accuracy(%) |
|---|----|----|----|----|----|----|--------------|
| 1 | 48 | 2 | 0 | 0 | 0 | 0 | 96 |
| 2 | 0 | 49 | 0 | 0 | 0 | 1 | 98 |
| 3 | 0 | 0 | 50 | 0 | 0 | 0 | 100 |
| 4 | 8 | 5 | 0 | 30 | 0 | 7 | 60 |
| 5 | 0 | 4 | 2 | 2 | 38 | 0 | 76 |
| 6 | 2 | 0 | 0 | 0 | 0 | 48 | 96 |

Overall Accuracy=90.751 %

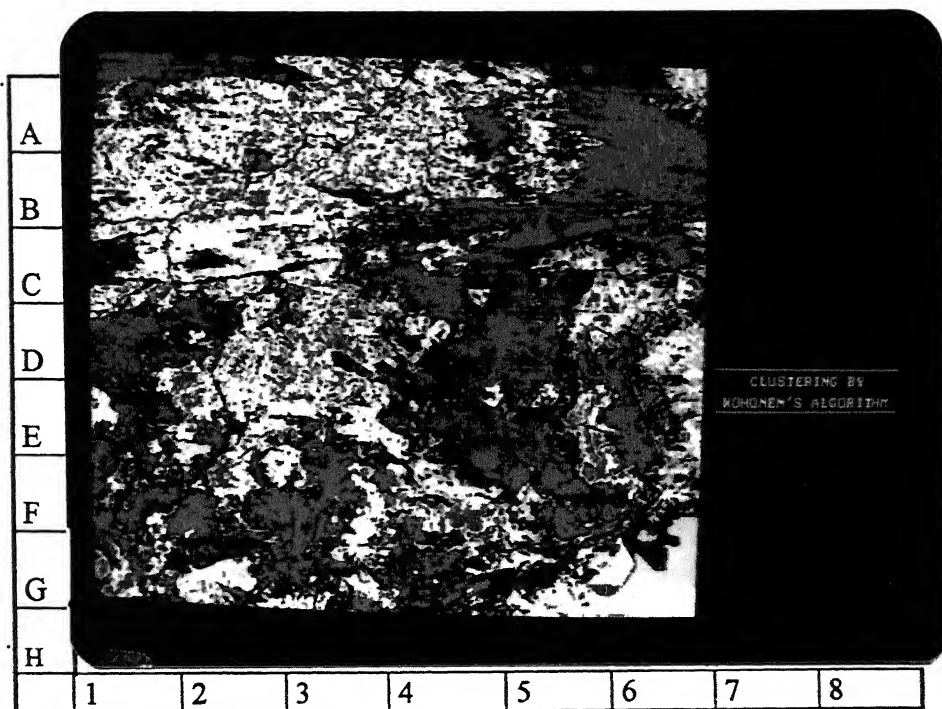


Fig 4.16 Classified image by Kohonen's algorithm

4.4.3 River extraction using Backpropagation learning

For this IRS -1B LISS II image of Kanpur displaying river Ganges is used. ANN was trained using Backpropagation learning having a desired output pattern as either 0 or 1 with hundred samples. The output value of 1 corresponded to the samples belonging to river (40 in number) while 0 corresponded to samples belonging to rest of the region. The samples from the entire image were selected such that all possible regions were present in the sample. The network required 5000 iterations to get trained. The image was then presented to the network. The output which was binary in nature having values of 0 or 1 was then displayed. This is shown in Fig. 4.17. River so extracted is shown in black, while the rest of the region is in white. There are, however, patches in black which are not a part of the river. These have resulted probably due to similar gray level values (i.e., similar to that of the river) of tributaries, and other surface drainage patterns present in the region.

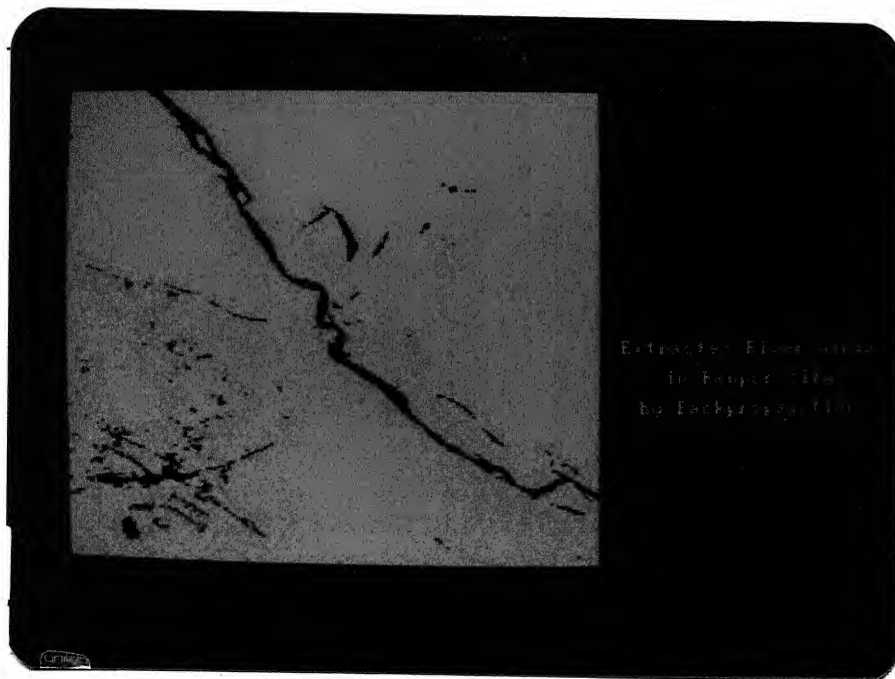


Fig. 4.17 River extracted using Backpropagation learning

Chapter Five

A Comparative Appraisal

5.0 General

The implementation of Neural Networks and Maximum likelihood classification was performed at the 'Agni' system of the HP-9000 series at IIT Kanpur. As earlier mentioned, all the algorithms were coded in C programming language by the author. In this work, two different ANN algorithms and one statistical pattern classification algorithm were investigated. The training/test data sets were kept same in all the algorithms thus facilitating comparison of the performance of each classifier. The performance has been assessed on the basis of the classification accuracy on the testing data set, time consumed by each classifier and the output classified image of the region.

5.1 Classification Accuracy

The accuracy attained in the maximum likelihood classification was 94.10 percent for the test area in level - I classification. The training of the classifier took virtually no time (of the order of 5-10 sec of user time). The input of training file was given sequentially and not randomly. Also the test area file had sequential inputs. This naturally improved the classifier performance. When the pixels from the image file were presented to the classifier, they were random in nature. The result of classification by MLC is shown in Fig.3.2. The region which are rocks has actually covered the sparse vegetation too. Similarly, the coal dumps have not come out so well at any locations. The reason could also be similarity in the mean vectors of these classes apart from the natural limitation of the classifier. The second algorithm used was the Backpropagation algorithm, which, essentially, is a supervised learning algorithm. This was tested for various iterations and the number of hidden layers was varied. Since, the learning rate parameter was kept as 0.8, the momentum factor was also kept high to prevent oscillation of the network. This

resulted in a faster convergence of the network and this was observed that number iteration necessary were also less to attain a high accuracy of 95.07 percent. Infact, the accuracy did not change between 500-10,000 iterations while it dropped at 20,000 iterations. It was also observed that on increasing the number of iterations, the efficiency of network dropped significantly and infact the network couldn't converge ever upto 20,000 iterations. The number of iterations required by one hidden layer networks was very small, to attain the above mentioned accuracy, this was primarily because, for remote sensing purposes where output is simply labels, the concentration is not on the mean square error but on the overall accuracy. The Table 4.1-4.6 summarise the results of the Back propagation algorithm for level-1 classifications. The classified image is shown in Fig.4.14. It has clearly brought out the clear water class and the turbid water class, while there has been a tremendous misclassification in the rocks class as the number of regions belonging to rocks have remained unclassified or wrongly classified. Rocks are shown in this image as yellow patches. Certain vegetation regions are also misclassified as rocks. The accuracy is , thus, less than that obtained in the test area data. This can be judged qualitatively on the basis of intensity of misclassification. The accuracy of classification by the Kohonen's algorithm is 90.75 percent. This is considerably less than the other two classifiers. The effect is seen visually in the image shown in Fig.4.16. The rocks have been classified to a very low accuracy, here they are indicated by purple colour, and have been classified as turbid water and coal dumps in most places. Similarly, the coal mines have been misclassified as vegetation, turbid water and coal dumps at number of places. A patch of clear water (indicated in light brown) is also misclassified as turbid water (shown as white).

In the level-II classification, the accuracy of MLC has increased marginally to 94.25 percent. However, mixing of rocks with vegetation is clearly seen in the image [Fig. 3.3]. It is also observed that lot of quartzite area is classified as the clear water. This is precisely because of presence of soil moist zones and ponds. Furthermore, the coal dumps and coal mines have come out reasonably well.

The level two classification taxed the efficiency of the Back propagation algorithm. As mentioned earlier, the learning rate parameter and the momentum factor values were reduced considerably. This resulted in an increase in the number of iterations. The network

showed a poor accuracy in class 3 which was vegetation category. This can be seen in the image of this classification. However, Coal mines, coal dumps, dense vegetation and water bodies were well classified. Sparse vegetation again got classified as rocks at a number of places.

Table 5.1 Accuracy(percent) summary of the three classifiers for level - I classification

| | Maximum Likelihood Classifier | Backpropagation algorithm | Kohonen's algorithm |
|-------------------------|--|--------------------------------------|--------------------------------|
| Clear water | 100 | 100 | 96 |
| Turbid water | 100 | 100 | 96 |
| Vegetation | 68 | 74 | 100 |
| Rocks | 96 | 100 | 60 |
| Coal mines | 100 | 100 | 76 |
| Coal dumps | 98 | 98 | 96 |
| Overall Accuracy | 94.10 | 95.07 | 90.75 |

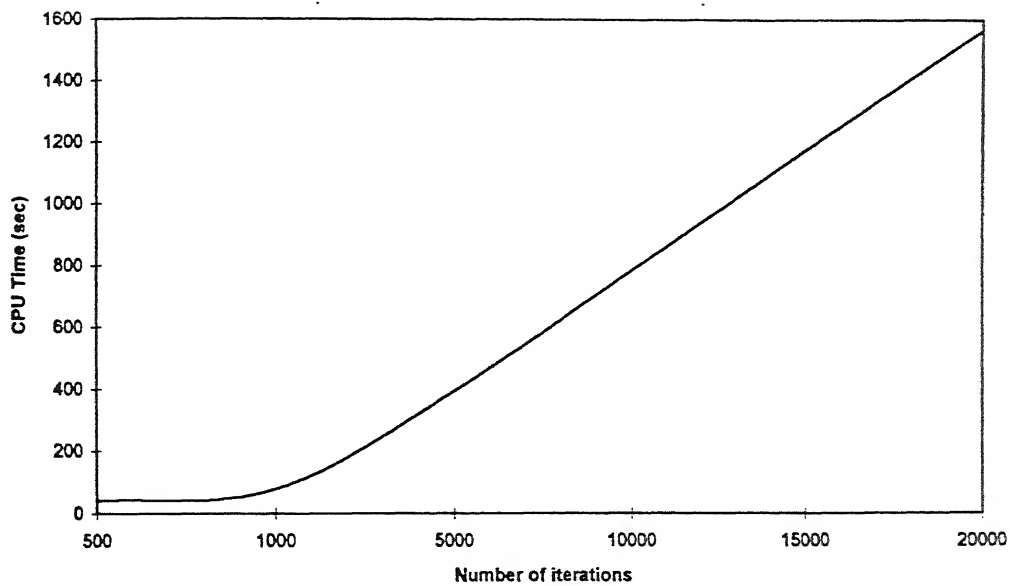
5.2 Computational time

The limitations of Neural Network is clearly spelt out when the cost of time is taken into consideration. Kohonen's clustering gave a good accuracy in very less time (150 sec, user time). However, it required lot of parameter to be selected before the final classification. The most important decision that had to be taken were of selection of the neighbourhood topology, selection of initial radius to select neighbours and gain term. Furthermore, dimensionality of our image often resulted in lot of trials in the values of gain term and initial radius.

The most time consuming algorithms was BP (Fig.5.1). This was because of need to select optimal network topology that could converge successfully. There were lot of parameters to be varied slowly to achieve the desired network topology. Furthermore, it is computationally more intensive hence, for large number of iterations, the execution went into hours. The data could not be fed as grey level values required normalisation. The

desired output pattern was to be decoded and recoded after the execution. Due to its computationally intensive nature, the training time ranged between 100s-1600s in the case of level-1 classification, while it varied between 150 sec - 6000 sec for level-2 classifications. The prediction time was, however, marginally less than MLE (84 sec).

Fig. 5.1 CPU Time vs Number of iterations



Conclusions and Future Recommendations

6.1 Conclusions

It was aimed in this study to evaluate the performance of ANN in pattern recognition and compare the results with the statistical Maximum Likelihood Classifier. The Maximum likelihood algorithm was chosen because it represents a widely used “standard” for comparison that yields minimum total classification error for Gaussian distributions. The conclusions drawn from this work are listed below:

- The primary computational difference between the algorithms is speed. The backpropagation approach to neural network training is extremely computation-intensive, taking considerably more time than the total classification time for the maximum likelihood. This is the most important drawback of the neural networks. However, the classification time, once the training is complete is less for the neural network.
- The Backpropagation algorithm could classify with higher accuracy in level - I as well as level -II classification than the Maximum likelihood.
- The Kohonen’s algorithm showed poorer classification accuracy.
- The results of the test data showed high accuracy while on visual analysis of the imagery, poorer classification results were obtained. This was true for all the classifiers. This shows that the accuracy reported, of any classification, depends quite a lot on the nature of training set as well as the test data. It was quite possible that certain regions of the imagery did not get represented well in the training and test data

set. Thus they were wrongly classified in the imagery. It was observed that the neural networks were more sensitive to the noise than the maximum likelihood classifier.

- The training procedure of neural network in this work was such that the inputs were not presented randomly to the network. Thus, when the image was fed as input, often wrong class assignments were observed. This could be another reason for lower accuracy of the images (qualitatively).
- It was observed while training the network using backpropagation learning that on increasing the number of layers the, network did not converge, hence one may safely conclude that smaller network topologies should be preferred to bigger networks.
- While selection of output representation of the backpropagation network, it was noticed that if the output was selected in single dimension, network could not converge well as the classes tended to 'pull' to each other depending on their real relationship. Though the fuzzy output of neural network is not directly related to the classification probability, clearly does depend on the likelihood of the different classes and their relationship with each other.
- This work has confirmed the belief that the neural networks being nonparametric are more robust to training site selection and class definition. The maximum likelihood on the other hand is more sensitive to the statistical considerations of the training sites. The classes are expected to follow Gaussian distribution pattern to give good results.
- In the backpropagation learning, for remote sensing purposes the final error (i.e., difference in the value of class label and the output) can be as high as 50% and still give excellent results at lower number of iterations.

- It was also seen in the present work that increase in number of classes from six to eight resulted in a complete change in the values of network parameters like learning rate and momentum parameter. This was because the dimensionality increased and hence the network begun oscillating and did not converge. Thus, smaller values of learning rate and momentum rates were to be used. Thus, on increasing the dimensionality of the network, it is advisable to slow down the learning process by using low learning rate and momentum rate values.
- Backpropagation algorithm was used to extract river in a scene. The results were very encouraging and demonstrated their usefulness in feature extraction. This could be extended to edge detection and several other feature extraction applications.

6.2 Recommendations for Future Work

1. It has been demonstrated by this work that the neural network is a useful tool for feature extraction. However, the biggest drawback is the large training times. Thus, implementation of the training algorithms on massively parallel computing systems can greatly enhance the applicability of the method.
2. The optimum number of training sets that can adequately represent the area, not studied in this work. This is an important feature of any classification especially when neural networks are known to work well with smaller training sets(Foody, 1995; Paola et al.,1995; Benediktsson, et al., 1993).
3. ANN may be used for edge detection for identifying geological structures such as faults and fractures.

References

- Aleksander, I., and H. Morton.. *An introduction to neural computing*. London: Chapman and Hall, 1990.
- Arbib, M.A. *Brains, Machines and Mathematics*, 2nd ed. NewYork : Springer and Verlag., 1987.
- Basu, T.N. and Srivastava, B.B.P. Structure and tectonics of Gondwana basins of peninsular India, *Prc. Fifth International Gondwana symposium*, Wellington, New Zealand.
- Benediktson, J.A., P.H. Swain and O.K., Ersoy. Neural network approaches versus statistical methods in classification of multisource remote sensing data, *IEEE Transaction on geoscience and remote sensing*, Vol. 28, No. 4, 540-552, 1990.
- Benediktson, J.A., P.H. Swain and O.K., Ersoy. Conjugate gradient neural networks in classification of multisource and very high dimensional remote sensing data. *International journal of remote sensing*, Vol. 14, No. 15, 2883-2903, 1993.
- Bolstad, P.V. and T.M., Lillesand. Rapid Maximum likelihood classification, Vol. 5. No. 1, *Photogrammetric Engineering and Remote Sensing*, 67-74, 1991.
- Churchland, P.S. and T.J., Sejnowsta. *The computational brain* : Cambridge MA : MIT Press, 1992.
- Duda, R.O. and P.E., Hart, *Pattern classification and scene analysis*: NewYork Wiley, 1973.
- Eggermount, J.J. *The correlative brain : Theory and experiment in neural interaction*. NewYork : Spinger-Verlag., 1990.
- E.R., Kaudel and Schwartz, *Principles of neural science*, Elsevier, NewYork , 1985.
- Faggin, F. VLSI implementation of neural networks, "Tutorial notes international joint conference on neural networks, Seattle WA., 1991.
- Foody, G.M., M.B., McCullods and W.B. Yates. Classification of remotely sensed data by an ANN : Issues related to training sets characteristics. *Photogrammetric engineering and remote sensing*, Vol. 61, No. 4, 391-401, 1995.

Foody, G.M. Using prior knowledge in artificial neural network classification with a minimal training set. *International journal of remote sensing*, Vol. 16, No. 2, 301-312, 1995.

Freeman, W.J. *Mass action in the nervous system.*, 1995.

Haykin S., *Neural Networks: A comprehensive foundation.* Macmillan College publishing company, New York, 1994.

Hepner, G.F., Logan, T., Ritter, N., and Bryant N. Artificial neural network classification using minimal training set : Comparison to conventional supervised classification. *Photogrammetric engineering and remote sensing*, 1990.

Joshi, K.C. and Pant, A. Coal resources of India, *Memoirs Geological Survey of India*, Vol. 8, pp. 291-298, 1971.

Kent, J.T., and Mardia, K.V. Spatial classification using fuzzy membership models. *IEEE Transaction on pattern recognition and machine intelligence*, 10, 659-671, 1988.

Lillesand, T.M. and Kiefer, R.W. *Remote sensing and image interpretation*, John Wiley & Sons, Inc., 1994.

Lippman, R.P "An introduction to computing with neural nets." *IEEE ASSP Magazine* 4, 4-22., 1987.

Mather, P.M. *Computer processing of remotely sensed images*, (Chichester : Wiley), 1987.

McCulloch, W.S. "A logical calculus of the ideas immanent in nervous activity." *Bulletin of biophysics*, 5, 45-133., 1943.

Mendel, J.M., and R.W., Mclarsen. 'Reinforcement learning control and pattern recognition systems." *Adaptive learning and pattern recognition systems : Theory and applications* (J.M. Mendel and K.S. Fu, ed.) , 1970.

Moon, W.Mon mathematical representation and integration of multiple spatial geoscience data sets. *Canadian journal of remote sensing*, 19, 63-67., 1993.

Murthy, M.V.N. some pebbles and boulders in Talcvhir Basal Conglomerates, Dudhi tehsil, mirzapur district, U.P., *Rec. Geol. surv. India*, 84(4), 1957.

Niblack, W. *Digital image processing*, PHI international., 1986.

Paola, J.P. and Schwengerdt R.A. A detailed comparison of Backpropagation neural network and Maximum likelihood classifiers for urban land classification, *IEEE Transaction on Geoscience and remote sensing*, Vol. 33, No. 4, 981-996, 1995.

Peddle, D.R. An empirical comparison of evidential reasoning, linear discrimination analysis and maximum likelihood algorithms for Alpine land cover classification. *Canadian journal of remote sensing*, 19, 31-44., 1993.

Raja Rao, C.S. (ed.) Coalfields of India, Vol. III, coal resources of M.P., J and K, *Bulletins of the Geological Survey of India*, 1983.

Richards, J.A. *Remote sensing digital image analysis*, Springer and Verlag., 1986.

Rosenblatt, F. "The perceptron : A probabilistic model for information storage and organization in the brain." *Psychological review* 65, 386-408., 1958.

Schalkoff, R.J. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc., 1992.

Schwengerdt, R.A.. *Techniques for image processing and classification in remote sensing*, Academic press, 249p, 1983.

Shepherd, G.M., and C., Koc. "Introduction of synaptic circuits." in *The synaptic organization of the brain* (G.M. Shepherd, ed.), pp. 3-31, New York : Oxford University press, 1990.

Swain, P.H. and Davis, S, *Remote sensing : A quantitative approach*, McGraw Hill Book Co., 396., 1978.

T., Kohonen, *Self organization and associative memory*, Springer - Verlag, Berlin, 1984.

Tripathi, N.K., "Classifiers in remote sensing - a comparative study", M. Tech. Thesis, I.I.T., Kanpur, 1987.

Tripathi, N.K. "Digital image processing of satellite data for structural mapping in Singrauli Coalfield", Ph. D. Thesis, I.I.T., Kanpur, 1994.

Venkateswarlu, N. B. "Evaluation of classifiers for pattern recognition", M. Tech. Thesis, I. I. T., Kanpur, 1988.

Wang, F. Fuzzy supervised classification of remote sensing images, *IEEE Transaction on geosciences and remote sensing*, 50, 193-207, 1990.

Wesserman, P.D. *Neural Computing : Theory and practice*. Van Nostrand Reinhold, New York.

121271

CE-1996-M-KAN-FEA



A121271
